GE Fanuc Automation

Automates Programmables Industriels

Module de communication CMM pour API 90–30

Manuel utilisateur

GFK–0582B–FR *Mai 1993*

Utilisation de l'expression "Attention danger" et des termes "Attention" et "Remarque" dans ce document

Attention danger

L'expression "Attention danger" est utilisée pour mettre en évidence des risques de blessures dues aux tensions, aux courants, aux températures ou à d'autres grandeurs physiques.

Toutes les situations où un manque d'attention peut être source de blessures physiques ou de dommages pour l'équipement sont repérées par cette expression.

Attention

Le terme "Attention" est associé aux situations où un manque d'attention risque de conduire à des dégâts matériels.

Remarque

Les "Remarques" ont pour but d'attirer votre attention sur des informations particulièrement utiles à la compréhension et à la mise en oeuvre de l'équipement.

Ce document est basé sur des informations disponibles au moment de sa publication. Malgré nos efforts de précision, nous ne pouvons prétendre couvrir tous les détails et toutes les variations matérielles ou logicielles possibles, ni aborder tous les cas de figure de l'installation, du fonctionnement ou de la maintenance. Les caractéristiques décrites dans ce document peuvent être absentes de certains systèmes matériels ou logiciels. GE Fanuc Automation ne s'engage pas à avertir les possesseurs de ce document d'éventuelles modifications ultérieures.

GE Fanuc Automation ne fournit aucune garantie explicite, implicite ou statutaire, et décline toute responsabilité quant à la précision, à l'utilité, et au caractère complet ou suffisant des informations contenues dans ce document. GE Fanuc Automation ne donne aucune garantie de qualité marchande et d'aptitude à une utilisation donnée.

Les marques suivantes sont des marques déposées de GE Fanuc Automation North America, Inc. :

Alarm Master	CIMSTAR	Helpmate	PROMACRO	Series Six
CIMPLICITY	GEnet	Logicmaster	Series One	Series 90
CIMPLICITY 90-ADS	Genius	Modelmaster	Series Three	VuMaster
CIMPLICITY PowerTRAC	Genius PowerTRAC	ProLoop	Series Five	Workmaster

Copyright 1989–1994 GE Fanuc Automation North America, Inc. Tous droits réservés

1. CONTENU DE CE MANUEL

Ce manuel décrit des produits de communication série pour Automates Programmables Industriels (API) Série 90. Toutes les informations nécessaires sont fournies pour vous permettre d'établir une liaison série entre un API Série 90, un ordinateur–hôte, des équipements périphériques ou un autre API.

Ce manuel se décompose comme suit :

Chapitre 1. Introduction : Explique les usages multiples du présent manuel. Décrit brièvement les produits de communication dont il fait l'objet.

Chapitre 2. Notions de communication série : Fournit une introduction à la transmission série principalement axée sur les communications avec des API Série 90.

Chapitre 3. Module CMM – Description, Installation et Configuration : Explique comment installer et configurer les modules coprocesseur de communication (CMM311, CMM711). Décrit les modules et la façon dont ils fonctionnent dans le système.

Chapitre 4. **Etablissement de la communication – instruction COMREQ :** Explique comment établir une communication à partir d'un programme à diagramme en échelle. La structure de l'instruction COMREQ est décrite en détail.

Chapitre 5. Service CCM : Définit les commandes du service CCM ; explique comment inclure les paramètres des commandes de ce service dans le Bloc de données de l'instruction COMREQ. De nombreux exemples de programmation à diagramme en échelle sont fournis.

Chapitre 6. Service SNP : Définit les commandes des services SNP et SNP–X ; explique comment inclure les paramètres des commandes de ces services dans le Bloc de données de l'instruction COMREQ. De nombreux exemples de programmation à diagramme en échelle sont fournis.

Chapitre 7. Définition des protocoles – CCM, RTU, SNP, SNP–X : Décrit les protocoles CCM, RTU, SNP, et SNP–X.

Chapitre 8. Schémas de câblage série : Explique comment fabriquer des câbles pour raccorder le module CMM à de nombreux autres équipements. Décrit le répéteur/convertisseur isolé.

Annexe A. Glossaire : Renferme une liste de termes concis, classés par ordre alphabétique se rapportant au vocabulaire des communications et qui fournit (si applicable) les acronymes qui leur sont associés.

Annexe B. Liste des codes ASCII : Fournit la table complète du jeu de caractères ASCII.

Annexe C. Mise en oeuvre CCM Série 90 : Décrit la mise en oeuvre du protocole CCM sur les API Série 90.

Annexe D. Mise en oeuvre RTU Série 90 : Décrit la mise en oeuvre du protocole RTU sur les API Série 90.

2. AUTRES MANUELS À CONSULTER

Pour plus de détails, se référer également aux ouvrages suivants :

- *GFK–0255 Series 90 Programmable Coprocessor Module and Support Software User's Manual*
- *GFK–0262 Series 90™–70 Programmable Controller Installation Manual*
- *GFK–0263 Logicmaster 90–70 Programming Software User's Manual*
- *GFK–0265 Series 90–70 Programmable Controller Reference Manual*
- *GFK–0356 Série 90™–30 Automate Programmable – Manuel d'installation*
- *GFK–0402 Series 90™–30 PLC Hand–Held Programmer User's Manual*
- *GFK–0466 Logicmaster™ 90–30/20/Micro Programming Software User's Manual*
- *GFK–0487 Series 90 PCM Development Software (PCOP) User's Manual*

3. VOS REMARQUES ET SUGGESTIONS SONT LES BIENVENUES

GE Fanuc Automation s'efforce d'éditer des documentations techniques de qualité. Après avoir utilisé ce manuel, merci de consacrer quelques instants à la page suivante, "Page de remarques", pour la compléter et nous la renvoyer.

Page de remarques

GFK–0582B–F	***Manuel utilisateur du module de communication CMM pour API 90–30***

Cochez votre fonction principale SVP

- ☐ Concepteur système
- ☐ Distributeur
- ☐ Intégrateur système
- ☐ Installateur
- ☐ Programmeur
- ☐ Responsable de maintenance
- ☐ Opérateur
- ☐ Autre (à préciser ci–dessous)

Si vous désirez une réponse personnelle, indiquez votre adresse postale complète :

SOCIETE : . . . . . NOM : . . . . . .

ADRESSE : . . . . . .

. . . . . PAYS : . . . . . .

Remettez cet imprimé directement à votre correspondant GE Fanuc ou envoyez–le à :

GE Fanuc Automation France
45, rue du Bois Chaland
CE 2904 – Lisses
91029 EVRY Cedex

Toutes vos remarques seront étudiées par du personnel qualifié.

REMARQUES

Si besoin, utilisez le verso de cette page.

Page de remarques

Sommaire

Page

CHAPITRE 1 – INTRODUCTION

1. RÉSUMÉ DU MANUEL 1–1

2. INTRODUCTION AUX MODULES COPROCESSEURS DE COMMUNICATION (CMM) 1–2
   - 2.1. Présentation du module CMM 1–2
   - 2.2. Partie matérielle du module CMM 1–4
   - 2.3. Configuration du module CMM 1–5
   - 2.4. Protocoles de communication du module CMM 1–5
     - 2.4.1. Protocoles SNP et SNP–X 1–5
     - 2.4.2. Protocole CCM 1–6
     - 2.4.3. Protocole RTU 1–6

CHAPITRE 2 – SÉRIE 90 : NOTIONS DE COMMUNICATION SÉRIE

1. PRINCIPE DES COMMUNICATIONS DU MODULE CMM 2–2
   - 1.1. Modèle de référence des communications du module CMM 2–2
   - 1.2. Description d'une communication avec le module CMM 2–3

2. L'INTERFACE SÉRIE 2–4
   - 2.1. Codes d'information 2–4
   - 2.2. Erreurs de transmission et Détection 2–5
     - 2.2.1. Erreurs de bruit 2–5
     - 2.2.2. Erreurs de synchronisation 2–6
   - 2.3. Transmission asynchrone 2–6
   - 2.4. Liaison série 2–7
     - 2.4.1. Modems 2–7
     - 2.4.2. Interfacesnormalisées 2–8

3. LE PROTOCOLE DE COMMUNICATION 2–10
   - 3.1. Protocoles de communication (SNP, SNP–X, CCM, RTU) 2–10
   - 3.2. Protocoles SNP et SNP–X 2–10
   - 3.3. Protocole CCM 2–11
   - 3.4. Protocole RTU 2–12

4. L'INTERFACE D'APPLICATION API 2–14

5. LES RÉSEAUX DE COMMUNICATION 2–15
   - 5.1. Point–à–point 2–15
   - 5.2. Multipoint 2–16
   - 5.3. Transmission par modem 2–17

Sommaire

CHAPITRE 3 – MODULE CMM – DESCRIPTION, INSTALLATION ET CONFIGURATION

1. DESCRIPTION MATÉRIELLE DU MODULE CMM ET FONCTIONNEMENT . . . 3–2
   - 1.1. Voyants . . . 3–3
     - 1.1.1. Voyant Module OK . . . 3–3
     - 1.1.2. Voyants Ports série . . . 3–3
   - 1.2. Bouton de Restart/Reset . . . 3–4
   - 1.3. Ports série . . . 3–4
     - 1.3.1. Ports série du module CMM711 . . . 3–4
     - 1.3.2. Ports série du module CMM311 . . . 3–5
   - 1.4. Fonctionnement système du module CMM . . . 3–7
     - 1.4.1. Fonctionnement système du module CMM Série 90–70 . . . 3–7
     - 1.4.2. Fonctionnement système du module CMM Série 90–30 . . . 3–8

2. INSTALLATION ET CONFIGURATION DU MODULE CMM . . . 3–9
   - 2.1. Configuration requise . . . 3–9
   - 2.2. Installation du module CMM . . . 3–10
     - 2.2.1. Généralités . . . 3–10
     - 2.2.2. Installation du module CMM dans le rack . . . 3–11
     - 2.2.3. Dépannage . . . 3–11
   - 2.3. Configuration du module CMM avec le logiciel Logicmaster 90 . . . 3–12
     - 2.3.1. Configuration du rack d'E/S . . . 3–12
     - 2.3.2. Configuration du module (configuration logicielle) . . . 3–15
   - 2.4. Configuration du module CMM311 avec la miniconsole de programmation HHP 3–25
     - 2.4.1. Gel de la configuration . . . 3–25
     - 2.4.2. Exemple de modification de la configuration d'un module CMM . . . 3–25

CHAPITRE 4 – ETABLISSEMENT DE LA COMMUNICATION – INSTRUCTION COMREQ

1. DEMANDE DE COMMUNICATION . . . 4–2
   - 1.1. Structure de la demande de communication . . . 4–2
   - 1.2. Principe de fonctionnement de la demande de communication . . . 4–3
   - 1.3. Synchronisation du traitement des demandes de communication . . . 4–4

2. INSTRUCTION COMREQ . . . 4–5

3. BLOC DE COMMANDE DE COMREQ . . . 4–7

CHAPITRE 5 – SERVICE CCM

1. LE BLOC DE DONNÉES CCM COMREQ . . . 5–2
   - 1.1. Structure du bloc de données CCM . . . 5–2
   - 1.2. Résumé du bloc de données pour les commandes CCM . . . 5–3
   - 1.3. Types de mémoire CCM . . . 5–5
     - 1.3.1. Zone de travail CCM (Type de mémoire 6) . . . 5–6
     - 1.3.2. Mots d'état de diagnostic . . . 5–8

1.4. Adressage de la mémoire CCM et longueur des données 5–9
1.4.1. Adresses mémoire CCM 5–9
1.4.2. Longueurs des données CCM 5–10

2. LE MOT D'ÉTAT CCM COMREQ 5–11

3. EXEMPLES DE PROGRAMMATION CCM COMREQ 5–14
3.1. Exemple de programme en diagramme en échelle 5–14
3.2. Exemples de commandes CCM 5–16
3.2.1. Commande Programmer la réponse Q : 06001 (1771) 5–16
3.2.2. Commande Effacer les mots d'état de diagnostic CCM : 06002 (1772) 5–17
3.2.3. Commande Lire les mots d'état de diagnostic CCM dans les registres Source : 06003 (1773) . . 5–17
3.2.4. Commande de configuration logicielle : 06004 (1774) 5–18
3.2.5. Commande Lire la mémoire de l'équipement destinataire dans la mémoire de l'équipement source : 06101–06103 (17D5–17D7) 5–19
3.2.6. Commande Lire la réponse Q dans la table des registres de l'équipement source : 06109 (17DD) 5–20
3.2.7. Commande Ecrire un seul bit : 06110 (17DE) 5–21
3.2.8. Commande Ecrire dans l'équipement destinataire depuis l'équipement source : 06111–06113 (17DF–17E1) 5–22

CHAPITRE 6 – SERVICE SNP

1. LE BLOC DE DONNÉES SNP COMREQ 6–2
1.1. Structure du bloc de données SNP 6–2
1.2. Types de mémoire SNP et adressage 6–3

2. LE MOT D'ÉTAT SNP 6–4
2.1. Codes d'erreurs majeures SNP 6–5
2.2. Codes d'erreurs mineures SNP 6–6

3. EXEMPLES DE PROGRAMMATION DE SNP COMREQ 6–17
3.1. Exemple de programme en diagramme en échelle 6–17
3.2. Exemples de commandes SNP 6–21
3.2.1. Commande Effacer Mots d'état de diagnostic : 07000 (1B58) 6–22
3.2.2. Commande Lire les mots d'état de diagnostic : 07001 (1B59) 6–23
3.2.3. Commande Modifier ID SNP : 07002 (1B5A) 6–25
3.2.4. Commande Programmer adresse des bits d'état X : 07003 (1B5B) 6–26
3.2.5. Commande Lire X : 07101 (1BBD) 6–28
3.2.6. Commande Ecrire X : 07102 (1BBE) 6–30
3.2.7. Commande de connexion : 07200 (1C20) 6–32
3.2.8. Commande Modifier le niveau de protection : 07201 (1C21) 6–36
3.2.9. Commande Lire Mémoire système : 07202 (1C22) 6–38
3.2.10. Commande Ecrire dans mémoire système : 07203 (1C23) 6–39
3.2.11. Commande Lire Mémoire des tâches : 07204 (1C24) 6–40
3.2.12. Commande Ecrire dans mémoire des tâches : 07205 (1C25) 6–41
3.2.13. Commande Lire Mémoire des blocs du programme : 07206 (1C26) 6–42
3.2.14. Commande Ecrire dans mémoire des blocs du programme : 07207 (1C27) 6–43
3.2.15. Commande Etat court de l'API : 07208 (1C28) 6–44
3.2.16. Commande Retourner Nom du programme de commande : 07209 (1C29) 6–46
3.2.17. Commande Retourner Type et ID du contrôleur : 07210 (1C2A) 6–48
3.2.18. Commande Retourner Heure/Date de l'API : 07211 (1C2B) 6–50
3.2.19. Commande Retourner Table des défauts : 07212 (1C2C) 6–51
3.2.20. Commande Programmer Heure/Date de l'API : 07213 (1C2D) 6–53
3.2.21. Commande Forcer bascule de la mémoire système : 07214 (1C2E) 6–54

3.2.22. Commande Etablir un datagramme : 07215 (1C2F) 6–56
3.2.23. Commande Mettre à jour le datagramme : 07216 (1C30) 6–59
3.2.24. Commande Annuler le datagramme : 07217 (1C31) 6–60
3.2.25. Commande Mettre à jour le datagramme en temps réel : 07218 (1C32) 6–61
3.2.26. Commande Connexion longue : 07300 (1C84) 6–63
3.2.27. Commande Numérotation automatique : 07400 (1CE8) 6–66

CHAPITRE 7 – DÉFINITION DES PROTOCOLES – CCM, RTU, SNP ET SNP–X

1. PROTOCOLE CCM 7–2
1.1. Présentation du protocole CCM 7–2
1.2. Mode CCM d'égal à égal 7–7
1.3. Mode CCM maître–esclave 7–9
1.4. Spécifications de synchronisation et du nombre d'essais 7–11
1.5. Tables des états du protocole CCM 7–13
1.5.1. Table des états CCM homologue 7–14
1.5.2. Actions de l'équipement CCM homologue 7–15
1.5.3. Table des états CCM maître 7–16
1.6. Actions de l'équipement CCM maître 7–17
1.6.1. Table des états CCM esclave 7–18
1.7. Actions de l'équipement CCM esclave 7–19

2. PROTOCOLE RTU 7–20
2.1. Introduction 7–20
2.2. Format des messages 7–20
2.3. Types de messages 7–21
2.4. Champs des messages 7–21
2.5. Format des caractères 7–23
2.6. Fin du message 7–23
2.7. Utilisation des temporisations 7–23
2.8. Contrôle de redondance cyclique (CRC) 7–24
2.8.1. Calcul du CRC–16 7–25
2.8.2. Exemple de calcul CRC–16 7–25
2.9. Calcul de la longueur de la trame 7–27
2.10. Description des messages 7–28
2.10.1. Erreurs de communication 7–43
2.10.2. Message d'interrogation invalide 7–43
2.10.3. Temporisations de la liaison série 7–44
2.10.4. Transactions invalides 7–45

3. PROTOCOLE SNP 7–46
3.1. Présentation du protocole SNP 7–46
3.2. Principe du mode SNP maître–esclave 7–46
3.3. Temporisateurs SNP 7–49
3.4. Datagrammes SNP 7–52

4. PROTOCOLE SNP–X 7–55
4.1. Présentation du protocole SNP–X 7–55
4.1.1. Session de communication SNP–X 7–55
4.1.2. Gestion des erreurs 7–56
4.1.3. Fonction de diffusion 7–56
4.1.4. Support des modems 7–56
4.1.5. Mot d'état de l'API esclave (état complémentaire) 7–57
4.1.6. Bits d'accès à la mémoire de l'équipement esclave (bits d'état X) 7–57
4.1.7. Temporisateurs 7–58
4.1.8. Break long 7–59
4.1.9. Code de contrôle de bloc (BCC) 7–60
4.2. Séquence du protocole SNP–X 7–61
4.2.1. Etablissement d'une session 7–62
4.2.2. Commandes directes 7–62
4.2.3. Commandes diffusées 7–63
4.3. Structure des messages SNP–X 7–65
4.3.1. Structure des messages de demande X 7–65
4.3.2. Structure des messages Réponse X 7–67
4.3.3. Structure du message Tampon X 7–69
4.4. Commandes SNP–X 7–70
4.4.1. Commande Connexion X 7–70
4.4.2. Commande Lire X 7–73
4.4.3. Commande Ecrire X 7–75
4.5. Tables des états SNP–X 7–82
4.5.1. Table des états de l'équipement SNP–X maître 7–82
4.5.2. Actions de l'équipement SNP–X maître 7–83
4.5.3. Table des états de l'équipement SNP–X esclave 7–84
4.5.4. Actions de l'équipement SNP–X esclave 7–85

CHAPITRE 8 – SCHÉMAS DE CÂBLAGE SÉRIE

1. SPÉCIFICATIONS D'ASSEMBLAGE DU CÂBLE 8–2

2. SCHÉMAS DE CÂBLAGE RS–232 8–3

3. SCHÉMAS DE CÂBLAGE RS–422/RS–485 8–4
3.1. Interface RS–422/RS–485 et schémas de câblage 8–4
3.2. schémas de câblage RS–422/RS–485 8–4

4. LE RÉPÉTEUR/CONVERTISSEUR ISOLÉ (IC655CCM590) 8–8
4.1. Description du répéteur/convertisseur isolé 8–8
4.2. Configurations système 8–12
4.3. Schémas de câblage 8–14

ANNEXE A – GLOSSAIRE

1. ACRONYMES ET ABRÉVIATIONS COURANTES A–2

2. GLOSSAIRE A–3

ANNEXE B – LISTE DES CODES ASCII

ANNEXE C – MISE EN OEUVRE CCM SÉRIE 90

ANNEXE D – MISE EN OEUVRE RTU SÉRIE 90

Tableaux

Page

Tableau 2-1. Format du code d'information ASCII 2–4
Tableau 2-2. Format des données série 2–6
Tableau 2-3. Signaux de l'interface normalisée (RS–232) 2–8

Tableau 3-1. Paramètres de communication CCM 3–21
Tableau 3-2. Paramètres de communication RTU 3–21
Tableau 3-3. Paramètres de communication SNP 3–21
Tableau 3-4. Temporisations SNP 3–24

Tableau 5-1. Résumé du bloc de commande COMREQ pour les commandes CCM 5–4
Tableau 5-2. Types de mémoire supportés par le protocole CCM Série 90 5–5
Tableau 5-3. Affectation de la mémoire de zone de travail CCM 5–6
Tableau 5-4. Définition des mots d'état de diagnostic CCM 5–8
Tableau 5-5. Adresses mémoire destinataire/source 5–9
Tableau 5-6. Unités de longueur des types de mémoire CCM Série 90 5–10
Tableau 5-7. Codes d'erreurs secondaires du CCM (Octet de poids fort du mot d'état COMREQ, Octet de poids faible de DSW) 5–12

Tableau 6-1. Types de mémoire, unités de longueurs et gammes valides 6–3
Tableau 6-2. Codes d'erreurs majeures du SNP 6–5
Tableau 6-3. Codes d'erreurs mineures correspondant au code d'erreur majeure 5 6–6
Tableau 6-4. Codes d'erreurs mineures correspondant au code d'erreur majeure 10 (0Ah) 6–7
Tableau 6-5. Codes d'erreurs mineures correspondant au code d'erreur majeure 12 (0Ch) 6–8
Tableau 6-6. Codes d'erreurs mineures correspondant au code d'erreur majeure 13 (0Dh) 6–12
Tableau 6-7. Codes d'erreurs mineures correspondant au code d'erreur majeure 14 (0Eh) 6–13
Tableau 6-8. Codes d'erreurs mineures correspondant au code d'erreur majeure 15 (0Fh) 6–14
Tableau 6-9. Commandes SNP 6–21
Tableau 6-10. Définition des mots d'état de diagnostic SNP 6–23
Tableau 6-11. Données d'état complémentaires 6–34
Tableau 6-12. Données du mot d'état de l'API esclave 6–35
Tableau 6-13. Description des niveaux de protection de l'API 6–37
Tableau 6-14. Niveaux de protection requis dans l'API 6–37

Tableau 7-1. Caractères de commande utilisés par le protocole CCM 7–3
Tableau 7-2. Description du bloc d'en–tête 7–4
Tableau 7-3. Types de mémoire Destinataire pour services CCM 7–5
Tableau 7-4. Description du bloc de données 7–7
Tableau 7-5. Délais d'attente de ENQ_ACK_TIMER 7–8
Tableau 7-6. Description de la demande ENQ (maître–esclave, Séquence normale) 7–9
Tableau 7-7. Description de la demande du maître (maître–esclave, Séquence Q) 7–11
Tableau 7-8. Description de la réponse de l'esclave (maître–esclave, Séquence Q) 7–11
Tableau 7-9. Temporisations du protocole CCM 7–12

Tableau 7-10. Nombre d'essais définis pour le protocole CCM 7–13
Tableau 7-11. Types de mémoire, unités de longueur et gammes valides des équipements esclaves . . . 7–23
Tableau 7-12. Longueur des messages RTU 7–27
Tableau 7-13. Affectation de la mémoire de zone de travail RTU 7–41

Tableau C–1. Commandes CCM Série 90 . . . . . C–1
Tableau C–2. Types de mémoire CCM supportés par les API Série 90 . . . . . C–2

Tableau D–1. Description des codes de fonction RTU D–1

Figures

Page

Figure 1-1. Module CMM Série 90–70 (CMM711) 1–3
Figure 1-2. Module CMM Série 90–30 (CMM311) 1–4

Figure 2-1. Modèle de référence CMM 2–2
Figure 2-2. Modems utilisés dans la liaison 2–7
Figure 2-3. Liaison RS–232 directe sans contrôle de flux 2–8
Figure 2-4. Liaison RS–232 avec modems sans contrôle de flux 2–9
Figure 2-5. Transferts de messages RTU 2–12
Figure 2-6. Exemple de configuration point–à–point RS–232 2–15
Figure 2-7. Exemple de configuration point–à–point RS–422/485 2–15
Figure 2-8. Exemple de configuration système multipoint 2–16
Figure 2-9. Exemple de configuration système avec transmission par modem 2–17

Figure 3-1. Module CMM Série 90–70 (CMM711) 3–2
Figure 3-2. Module CMM Série 90–30 (CMM311) 3–3
Figure 3-3. Fonctions affectées aux broches des ports série du module CMM Série 90–70 3–4
Figure 3-4. Fonctions affectées aux broches des ports série du module CMM Série 90–30 3–5
Figure 3-5. Branchements du câble en ”Y” du module CMM Série 90–30 3–6
Figure 3-6. Schéma de principe du module CMM Série 90–70 3–7
Figure 3-7. Schéma de principe du module CMM Série 90–30 3–8
Figure 3-8. Configurations d’un module CMM Série 90–70 3–10
Figure 3-9. Contrl Flux = AUCUN (ou SANS), DELAI TurnA=0 3–22
Figure 3-10. Contrl Flux = AUCUN (ou SANS), DELAI TurnA=100 ms 3–22
Figure 3-11. Contrl Flux = MATERIEL 3–23

Figure 4-1. Structure de l’instruction COMREQ 4–2
Figure 4-2. Principe de fonctionnement de la demande de communication 4–3

Figure 7-1. Format du bloc de données 7–6
Figure 7-2. Transfert de données de Source vers Destinataire (Ecriture, égal à égal) 7–7
Figure 7-3. Transfert de données de Destinataire vers Source (Lecture, égal à égal) 7–8
Figure 7-4. Format de la demande ENQ (maître–esclave, Séquence normale) 7–9
Figure 7-5. Transfert de données du maître vers l’esclave (Ecriture, séquence maître–esclave normale) . . . 7–10
Figure 7-6. Transfert de données de l’esclave vers le maître (Lecture, séquence maître–esclave normale) . 7–10
Figure 7-7. Transfert de données de l’esclave vers le maître (Séquence Q maître–esclave) 7–10
Figure 7-8. Transferts des messages RTU 7–20
Figure 7-9. Registre de Contrôle de Redondance Cyclique (CRC) 7–24
Figure 7-10. Flux des messages : Etablir une session de communication 7–48
Figure 7-11. Flux des messages : Demande et réponse 7–48
Figure 7-12. Flux des messages : Mettre à jour le datagramme en temps réel 7–53
Figure 8-1. Module CMM à module CMM avec établissement d’une liaison (RS–232 seulement) 8–3
Figure 8-2. Module CMM à Terminal Opérateur d’Interface avec établissement d’une liaison (RS–232) . . 8–3

Figures

Figure 8-3. Module CMM à Terminal Opérateur d'Interface sans établissement d'une liaison (RS–232) . . . 8–3
Figure 8-4. Module CMM à module CMM avec Contrl Flux = Aucun (RS–422/RS–485) 8–5
Figure 8-5. Module CMM à module CMM avec Contrl Flux = Matériel (RS–422/RS–485) 8–5
Figure 8-6. Module CMM à Terminal Opérateur d'Interface avec Contrl Flux = Aucun (RS–422/RS–485) 8–5
Figure 8-7. Module CMM ou ordinateur–hôte à plusieurs modules CMM (Multipoint 4 fils) 8–6
Figure 8-8. Module CMM à module CMM avec Contrl Flux = Aucun (RS–422/RS–485 2 fils) 8–6
Figure 8-9. Module CMM à plusieurs modules CMM (Multipoint 2 fils) 8–7
Figure 8-10. Répéteur/convertisseur isolé 8–9
Figure 8-11. Schéma de principe du répéteur RS–422/convertisseur RS–232 isolé 8–10
Figure 8-12. Configuration système simple avec un seul répéteur/convertisseur isolé 8–13
Figure 8-13. Configuration système complexe avec plusieurs répéteurs/convertisseurs isolés 8–13
Figure 8-14. Câble A ; CMM RS–232 à convertisseur 8–14
Figure 8-15. Câble B ; CMM RS–422 à convertisseur 8–14
Figure 8-16. Câble C ; Paire torsadée RS422 8–15
Figure 8-17. Câble D ; Paire torsadée RS422 8–16
Figure 8-18. Câble E ; Convertisseur RS–232 à CMM 8–16

Chapitre 1 Introduction

Ce chapitre renferme un résumé du manuel et présente brièvement les modules coprocesseurs de communication.

1. RÉSUMÉ DU MANUEL

Ce manuel a pour principal objectif de décrire l'utilisation des modules coprocesseurs de communication (CMM) tant du point de vue matériel que du point de vue logiciel. La partie concernant le matériel explique comment installer les modules, fabriquer et connecter les câbles de communication série. La partie concernant le logiciel explique comment configurer les modules à l'aide du logiciel Logicmaster 90 et programmer les demandes de communication série dans des programmes à diagramme en échelle. Et pour les utilisateurs avertis, les protocoles de communication qui résident dans les modules CMM sont décrits en détail.

Ce manuel a pour autre objectif de rappeler certaines notions élémentaires en matière de transmission de données série en rapport direct avec les produits de communication série de GE Fanuc Automation.

Ce manuel couvre ainsi un nombre important de sujets que la plupart des utilisateurs n'auront pas à étudier dans leur intégralité. Le tableau ci–dessous a été conçu pour vous aider à accéder rapidement aux informations dont vous avez besoin.

Sujets	Place dans le manuel	
Notions de communications	Chapitre 1 :	Introduction
	Chapitre 2 :	Notions de transmission de données
Matériel		
– Installation du module	Chapitre 3 :	Module CMM – Description, Installation, et Configuration
– Câbles de communication, Répéteur/Convertisseur isolé	Chapitre 8 :	Schémas des câbles série
Logiciel		
– Configuration du module	Chapitre 3 :	Module CMM – Description, Installation, et Configuration
– Programmation à diagramme en échelle	Chapitre 4 :	Etablissement de la communication – instruction COMREQ
	Chapitre 5 :	Service CCM
	Chapitre 6 :	Service SNP
– Description des protocoles	Chapitre 7 :	Définition des protocoles – CCM, RTU, SNP, SNP–X

2. INTRODUCTION AUX MODULES COPROCESSEURS DE COMMUNICATION (CMM)

Dans le présent manuel, le module coprocesseur de communication est désigné par ”CMM”, abréviation de la référence produit du module. Les modules CMM suivants sont décrits dans ce manuel :

- API Série 90–70 : IC697CMM711
- API Série 90–30 : IC693CMM311

Les modules CMM Série 90–70 et Série 90–30 ont des fonctions et un mode opératoire très similaires ; c’est pourquoi ils sont décrits en même temps dans la plupart des cas.

Remarque

Veillez à ne pas confondre le terme ”CMM” (module coprocesseur de communication) et le terme ”CCM” (l’un des protocoles de communication série des modules CMM).

2.1. PRÉSENTATION DU MODULE CMM

Le module coprocesseur de communication (CMM) est un coprocesseur hautes performances conçu pour gérer la partie communication dans un système d’API Série 90™.

Le module CMM est étroitement lié à l’API Série 90 et peut être configuré pour fournir deux ports de communication indépendants. Dans de nombreuses applications, chaque port fonctionne comme une fenêtre indépendante de l’API à laquelle d’autres équipements tels que des calculateurs industriels et des terminaux graphiques couleur peuvent accéder. La plupart des applications qui auparavant accédaient à l’API Série 90 via le port série interne de la CPU de l’API par le protocole SNP, peuvent maintenant accéder directement à l’API via le module CMM.

Chaque module CMM occupe un seul emplacement dans le rack d’un API Série 90. Jusqu’à 63 modules CMM peuvent être installés dans un seul système Série 90–70 pour améliorer l’accès aux équipements d’E/S série et accéder à la mémoire des API. Dans un API Série 90–30, jusqu’à 9 modules CMM peuvent être installés dans le rack principal.

Le module CMM Série 90–70 se configure à l’aide du logiciel de configuration Logicmaster 90–70. Le module CMM Série 90–30 peut être configuré soit par le logiciel Logicmaster 90, soit par la miniconsole de programmation HHP.

Les schémas ci–dessous montrent la façade avant des modules et la position des principales fonctions sur les cartes :

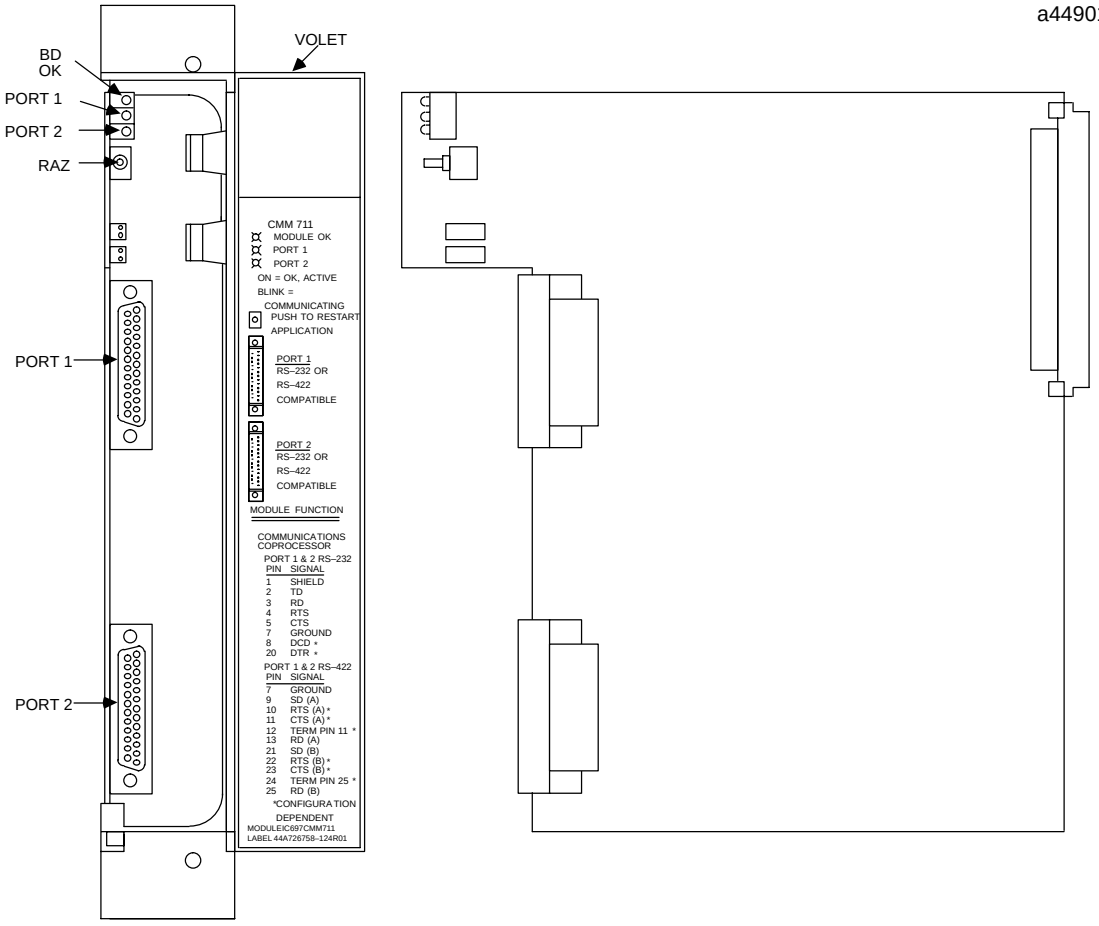


Figure 1-1. Module CMM Série 90–70 (CMM711)

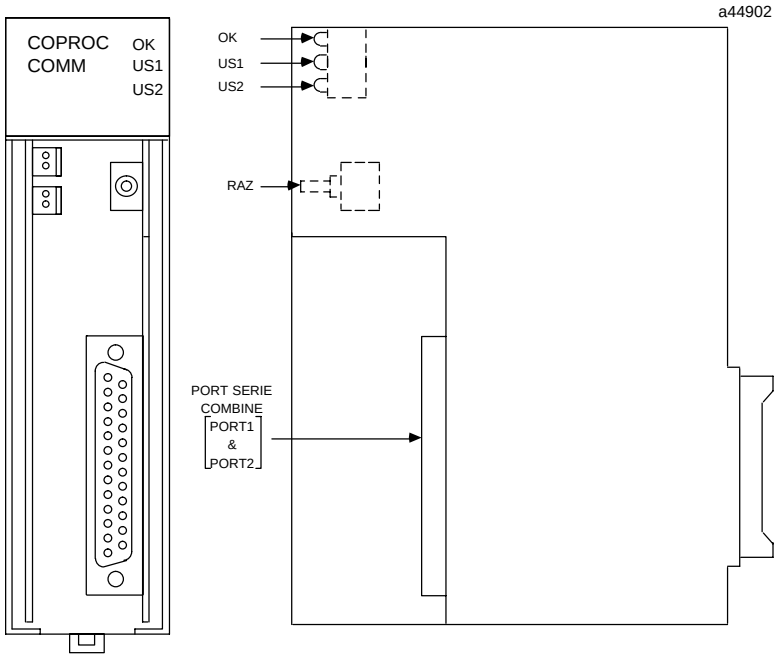


Figure 1-2. Module CMM Série 90–30 (CMM311)

2.2. PARTIE MATÉRIELLE DU MODULE CMM

La partie matérielle d'un module CMM Série 90 se décompose comme suit :

Ports série : Deux ports sont fournis pour des interfaces série avec des écrans, des lecteurs de codes–barres et autres équipements. Ces ports fonctionnent de la même manière et peuvent être configurés de façon logicielle en mode RS–232 et/ou RS–485/RS–422.

Sur le module CMM Série 90–30, la connexion sur les deux ports utilise un seul connecteur 25 broches type D. Un câble en "Y" est fourni avec chaque module CMM Série 90–30. Ce câble de 0,3 m est équipé d'un connecteur coudé à angle droit à une extrémité pour le raccordement à la carte CMM. L'autre extrémité est équipée d'un connecteur double, un connecteur pour le port 1 et l'autre pour le port 2.

Voyants : Le module CMM est équipé de trois voyants qui vous permettent de connaître l'état du module CMM sans avoir à raccorder un terminal. Le voyant MODULE OK (voyant du haut) indique l'état courant de la carte CMM. Les voyants PORT1 (US1) et PORT2 (US2) (voyants du milieu et du bas, respectivement) vous renseignent sur l'activité en émission et en réception des ports série 1 et 2, respectivement.

Bouton de RAZ/Reset : Le module CMM dispose également d'un bouton de RAZ/Reset qui permet de réinitialiser la carte à un état connu.

Si vous appuyez sur le bouton de RAZ/Reset, le module CMM sera réinitialisé et relancé à partir des données de la configuration logicielle utilisateur. Une remise sous tension forcera également le logiciel à redémarrer en mode de configuration utilisateur. Ce mode est le mode de fonctionnement normal du module CMM, une fois qu'il a été programmé et configuré.

2.3. CONFIGURATION DU MODULE CMM

Lorsque vous recevez le module CMM, vous devez le configurer pour spécifier les fonctions de communication qu'il devra remplir.

Avec un module CMM Série 90–70, vous devez utiliser le logiciel de configuration Logicmaster 90 pour configurer l'API pour le module CMM. Avec un module CMM Série 90–30, vous pouvez utiliser soit le logiciel de configuration Logicmaster 90, soit une miniconsole de programmation (HHP) pour configurer l'API pour le module CMM.

Le logiciel de configuration Logicmaster 90 est accepté par de nombreux ordinateurs. Pour plus d'informations à ce sujet, voir les manuels d'utilisation Logicmaster 90.

2.4. PROTOCOLES DE COMMUNICATION DU MODULE CMM

2.4.1. Protocoles SNP et SNP–X

Les protocoles SNP et SNP–X sont fournis en EPROM pour les modules CMM Série 90–70 comme pour les modules CMM Série 90–30. Le protocole SNP est un protocole de communication propriétaire. C'est le protocole de communication natif de tous les API de la gamme Série 90.

Le protocole SNP–X est une extension optimisée du SNP. S'il comporte moins de fonctions que SNP, il est plus simple à utiliser et offre surtout un niveau de performance nettement amélioré. Le protocole SNP–X ne peut être installé que lorsque le protocole SNP est déjà configuré et opérationnel.

Les protocoles SNP et SNP–X permettent les modes de fonctionnement suivants :

- Maître – Equipement initiateur dans un système maître/esclave.
- Esclave – Equipement répondeur dans un système maître/esclave.

Le protocole SNP maître–esclave tel qu'il est mis en oeuvre dans le module CMM ne supporte pas les fonctions de programmation ou de configuration de l'API. Tout logiciel Logicmaster 90 en liaison avec le port série d'un module CMM configuré comme un équipement SNP esclave ne peut être utilisé qu'à des fins d'affichage et de modification des données. (Le protocole SNP–X ne supporte en aucun cas les fonctions de programmation ou de configuration de l'API).

Une fois configurés, les protocoles SNP et SNP–X sont validés sur le ou les ports sélectionnés. Ils peuvent être validés sur aucun, un seul ou les deux ports série du module CMM en mode RS–232 ou RS–422/RS–485. Toute combinaison de protocoles, ports et normes électriques est acceptée, à une seule exception près : le module CMM Série 90–30 **ne supporte pas** l'interface RS–422/RS–485 sur le port 1. Le(s) port(s), la vitesse, la parité, le contrôle de flux, le nombre de bits de stop, les temporisations et les temps de retournement peuvent être définis par l'utilisateur.

2.4.2. Protocole CCM

Le protocole CCM est fourni en EPROM pour les modules CMM Série 90–70 comme pour les modules CMM Série 90–30.

Le protocole CCM peut être configuré comme suit :

- Maître – Equipement initiateur dans un système maître/esclave.
- Esclave – Equipement répondeur dans un système maître/esclave.
- Egal à égal – Etablit la communication et répond à un autre équipement en mode égal à égal.

Une fois configuré, le protocole CCM est validé sur le ou les ports sélectionnés. Il peut être validé sur aucun, un seul ou les deux ports série du module CMM en mode RS–232 ou RS–422/RS–485. Toute combinaison de protocoles, ports et normes électriques est acceptée, à une seule exception près : le module CMM Série 90–30 **ne supporte pas** l'interface RS–422/RS–485 sur le port 1. Le(s) port(s), la vitesse, la parité, le contrôle de flux, les temporisations, les temps de retournement et le nombre d'essais peuvent être définis par l'utilisateur.

2.4.3. Protocole RTU

Le protocole RTU est fourni en EPROM pour les modules CMM Série 90–70 comme pour les modules CMM Série 90–30. Le protocole RTU mis en oeuvre dans le module CMM est un sous–ensemble du protocole de communication série Modbus Remote Terminal Unit.

Le protocole RTU tel qu'il est mis en oeuvre dans le module CMM ne fonctionne qu'en mode esclave.

- Esclave – Equipement répondeur dans un système maître/esclave.

Une fois configuré, le protocole RTU est validé sur le ou les ports sélectionnés. Il peut être validé sur aucun, un seul ou les deux ports série du module CMM en mode RS–232 ou RS–422/RS–485. Toute combinaison de protocoles, ports et normes électriques est acceptée, à une seule exception près : le module CMM Série 90–30 **ne supporte pas** l'interface RS–422/RS–485 sur le port 1. Le(s) port(s), l'adresse de la station, la vitesse, le contrôle de flux et la parité peuvent être définis par l'utilisateur.

Chapitre 2 Série 90 : notions de communication série

Ce chapitre décrit brièvement le principe de la communication série. Le terme de communication série recouvre de nombreux aspects tels que les programmes d'application, les protocoles de communication et les interfaces matérielles. A titre pratique, un modèle de communication série est fourni pour vous permettre de comprendre l'architecture de la communication série sous module CMM. Le modèle proposé est décrit en détail dans les différents paragraphes de ce chapitre.

Le présent chapitre traite des sujets suivants :

- 1. Principe des communications du module CMM
- 2. L'interface série
- 3. Le protocole de communication
- 4. L'interface d'application API
- 5. Les réseaux de communication

Pour bien comprendre, le lecteur doit connaître les systèmes de numérotation binaire et hexadécimale et avoir assimilé le fonctionnement d'un API (Automate Programmable Industriel). Les informations contenues dans ce chapitre ne sont fournies qu'à titre indicatif. Des informations plus spécifiques sur les modules coprocesseurs de communication pour API Série 90 et tous les sujets qui s'y rapportent figurent dans les chapitres qui suivent.

1. PRINCIPE DES COMMUNICATIONS DU MODULE CMM

Ce paragraphe présente un modèle de référence des communications du module CMM. Un exemple de transfert d'un message CMM typique est également fourni pour clarifier les concepts du modèle de référence.

1.1. MODÈLE DE RÉFÉRENCE DES COMMUNICATIONS DU MODULE CMM

Ce paragraphe illustre et décrit brièvement les fonctions de communication du module CMM à partir du modèle de référence CMM fourni. Le schéma ci–dessous illustre les principales fonctions de communication du module CMM entre un ordinateur–hôte et un API Série 90 ou entre deux API Série 90 :

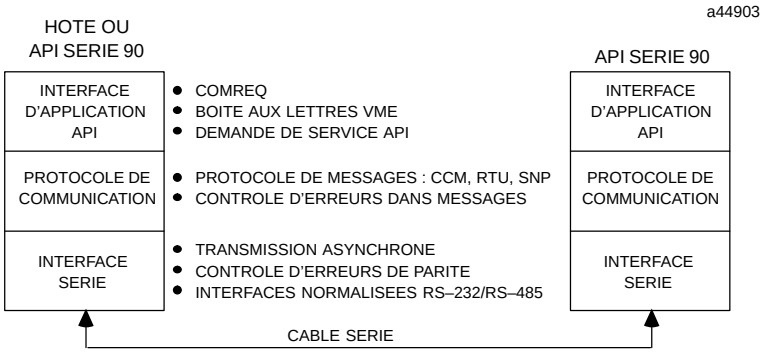


Figure 2-1. Modèle de référence CMM

Interface série : L'interface série code les caractères des messages, fournit la transmission électrique des messages sur la liaison série et contrôle la parité des caractères entrants.

Protocole de communication : Le protocole de communication définit la séquence d'établissement et d'interruption des liaisons pour l'envoi de messages et le transfert des données. Le module CMM dispose des protocoles SNP, SNP–X, CCM et RTU. (Ces protocoles sont décrits en détail dans le Chapitre 7). Le protocole de communication est également chargé du contrôle d'erreurs au niveau des messages.

Interface d'application API : L' interface d'application API est l'interface de programmation qui permet d'établir les communications série sur le module CMM. L'API Série 90 utilise l'instruction COMREQ du programme utilisateur pour transférer des informations entre la CPU de l'API et le module CMM.

1.2. DESCRIPTION D'UNE COMMUNICATION AVEC LE MODULE CMM

Les paragraphes qui suivent expliquent ce qui se passe lors du transfert d'un message entre un API Série 90 source (initiateur) et un API Série 90 destinataire (répondeur). Le cheminement du message est décrit depuis le programme utilisateur de l'équipement source, via le modèle CMM puis la liaison série jusqu'à l'équipement destinataire. L'exemple explique ce qui se passe lorsque le protocole CCM est utilisé. Un "message" peut être défini ici comme une demande de transfert de données, le transfert en soi et toute indication relative à l'état du transfert.

Etablissement de la communication. Le transfert d'un message entre un API Série 90 source et un API Série 90 destinataire est déclenché par une instruction COMREQ dans le programme utilisateur. L'instruction COMREQ sert de pont entre le programme utilisateur et l'interface d'application API du module CMM. Lorsqu'une instruction COMREQ est générée, des informations spécifiques décrivant le message sont transférées dans l'environnement du module CMM. L'interface d'application API est chargée de transmettre les détails du message au protocole de communication, d'assurer le transfert des données entre la CPU de l'API et le module CMM, et de mettre à jour l'état du message dans la CPU de l'API.

Conversion du message au format du protocole. Lorsque le protocole de communication reçoit le message de l'interface d'application API, il code les détails du message sous un format qui lui est propre (format du protocole CCM en l'occurrence). Le format regroupe les détails du message et les données nécessaires à l'identification de l'équipement destinataire au niveau des messages. Le message est alors transmis à l'interface série.

Transmission du message sur la liaison série. L'interface série prépare le message au format du protocole pour pouvoir le transmettre sur la liaison série. Elle code, entre autres, chaque caractère du message en fonction d'un code d'information standard. L'interface série fournit également les gestionnaires électriques des supports physiques par lesquels transiteront les données codées. Pour la communication avec le module CMM, il existe deux types de gestionnaires : le premier basé sur la norme RS–232, le second sur les normes RS–422/RS–485.

Action entreprise par l'équipement destinataire. Lorsque l'équipement destinataire reçoit le message, il commence par décoder tous les caractères entrants. Le protocole de communication de l'équipement destinataire vérifie la destination et décode les autres détails du message. Si le message demande à l'équipement destinataire des données pour lecture, l'équipement destinataire reçoit les données de sa CPU via l'interface d'application API, code les données au niveau du protocole et des caractères et les renvoie à l'équipement source. Si le message demande l'écriture de données dans l'équipement destinataire, les données transmises par l'équipement source sont alors communiquées à la CPU de l'API destinataire, via l'interface d'application API.

Au fur et à mesure que les caractères sont reçus par l'équipement source ou destinataire, ils subissent un contrôle d'erreurs à la fois au niveau du protocole de communication et de l'interface série. A la fin du transfert de données, la connexion est libérée. De plus, l'interface d'application API de l'équipement source renvoie l'état du transfert à la CPU de l'API.

2. L'INTERFACE SÉRIE

L'interface série code/décode les messages en fonction d'un code d'information particulier, et contrôle la parité de chaque caractère reçu sur la liaison série. Les gestionnaires de la ligne de transmission sont également fournis par l'interface série.

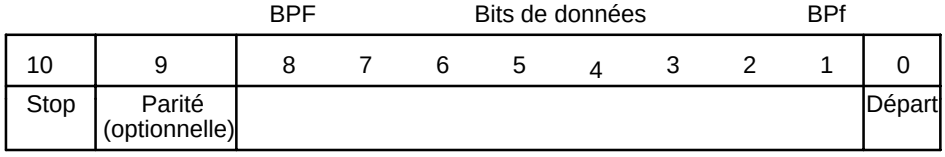
Ce paragraphe traite des sujets suivants :

- Codes d'information
- Erreurs de transmission et détection
- Transmission asynchrone
- Normes de l'interface série (RS–232, RS–422, RS–485)

2.1. CODES D'INFORMATION

Un code d'information est une convention de représentation des nombres, des lettres, des symboles et des caractères de commande utilisée en transmission série. Dans le protocole CCM, les caractères dans les en–têtes et les caractères de commande sont codés. Les autres caractères tels que ceux qui figurent dans les données, sont des données binaires non codées. Il existe un certain nombre de schémas de codage à ce jour, mais le plus courant est celui utilisé dans les communications du module CMM à savoir le code ASCII (American Standard Code for Information Interchange).

Comme le montre le schéma ci–après, le module CMM utilise un codage de caractère sur 8 bits, plus un bit de parité optionnel pour le transfert des données série.



Le tableau ci–après montre des exemples au format binaire et hexadécimal, avec bit de parité, de plusieurs caractères ASCII. Le bit de parité est décrit dans le paragraphe intitulé Contrôle de parité. La liste complète du jeu de caractères ASCII en hexadécimal et en décimal est fournie en Annexe B.

Tableau 2-1. Format du code d'information ASCII

Bit de parité	Format binaire du caractère	Format hexadécimal du caractère	Caractère ASCII
(impaire) 0	0 0 0 0 0 0 1 0	0 2	STX (caractère de commande) Début de texte
(impaire) 1	0 0 1 0 1 0 1 1	2 B	+
(paire) 1	0 0 0 1 0 1 0 1	1 5	NAK (caractère de commande) Accusé de réception négatif
(paire) 0	0 0 1 1 1 0 0 1	3 9	9

2.2. ERREURS DE TRANSMISSION ET DÉTECTION

Tous les protocoles série asynchrones utilisent des mécanismes de contrôle et de détection d'erreurs pour réduire le nombre d'erreurs de transmission et garantir un transfert fiable des données. Les méthodes de contrôle d'erreurs utilisées par les protocoles de communication du module CMM sont décrites ci–dessous.

2.2.1. Erreurs de bruit

Le module CMM utilise quatre types de contrôles d'erreurs de bruit :

- Contrôle de parité.
- Contrôle du code de contrôle de bloc BCC (SNP, SNP–X).
- Contrôle de redondance longitudinale LRC (CCM).
- Contrôle de redondance cyclique CRC (RTU).

Le contrôle du code de contrôle de bloc BCC (protocoles SNP, SNP–X), le contrôle de redondance longitudinale LRC (protocole CCM), et le contrôle de redondance cyclique CRC (protocole RTU) sont effectués dans le protocole de communication et sont décrits au paragraphe 3 de ce chapitre.

2.2.1.1. Contrôle de parité

En règle générale, le contrôle de parité est défini à Sans, Paire ou Impaire. Le bit de parité, calculé par l'émetteur et contrôlé par le récepteur, dépend du nombre de "1" dans le caractère codé en binaire. Si la parité est définie à impaire, le nombre total de "1" dans le caractère codé en binaire (en plus du bit de parité) doit être impair. Si la parité est paire, le nombre total de "1" dans le caractère, y compris le bit de parité, doit être pair. Si la parité a été définie à Sans, aucun contrôle de parité n'est effectué.

Dans l'exemple ci–dessous, le caractère "A" codé en ASCII renferme deux "1", ce qui signifie que le bit de parité doit être à "1" pour une parité impaire. Le bit de parité devrait être à "0" si la parité avait été définie à paire. Lorsque la parité est définie à Sans, le bit de parité n'est pas transmis. Pour le protocole CCM, le bit de parité optionnel peut être défini à Pair ou Sans ; pour les protocoles SNP, SNP–X et RTU, la parité peut être définie à Impaire, Paire ou Sans.

Si le contrôle de parité est utilisé et que l'un des bits est transmis de façon incorrecte, le bit de parité reflètera l'erreur.

2.2.1.2. Caractère ASCII, "A", reçu correctement

Bit de parité	Octet de données reçu							
(impaire)	8	7	6	5	4	3	2	1
1	0	1	0	0	0	0	0	1

2.2.1.3. Caractère ASCII, "A", reçu avec une erreur dans le premier bit

Bit de parité	Octet de données reçu							
(impaire)	8	7	6	5	4	3	2	1
1	0	1	0	0	0	0	0	0

Le récepteur contrôle le bit de parité et détecte l'erreur de transmission : le caractère reçu (bit de parité compris) comporte un nombre pair de "1" au lieu d'un nombre impair.

Cependant, si le nombre de bits transmis incorrectement dans un caractère est pair, le bit de parité ne reflètera pas l'erreur.

2.2.1.4. Caractère ASCII, "A", reçu avec des erreurs dans les deux premiers bits

Bit de parité	Octet de données reçu							
(impaire)	8	7	6	5	4	3	2	1
1	0	1	0	0	0	0	1	0

Le bit de parité ne reflète pas l'erreur étant donné que le caractère reçu (bit de parité compris) comporte un nombre impair de "1", comme il se doit.

2.2.2. Erreurs de synchronisation

Une mauvaise synchronisation entre émetteur et récepteur génère d'autres types d'erreurs tels que des erreurs de cadence, de perte d'encadrement de trame et de temporisation.

Erreurs de cadence. Si par suite d'une mauvaise synchronisation entre l'émetteur et le récepteur les caractères sont envoyés plus vite que ne peut les traiter le récepteur, on aboutit à une situation connue sous le nom d'erreur de cadence. Dans ce cas, le caractère précédent est remplacé par le suivant et une erreur est signalée.

Erreurs de perte d'encadrement de trame. En transmission asynchrone, ce type d'erreur se produit lorsque le récepteur confond un bit de données 0 logique ou une pointe de bruit avec un bit de départ. L'erreur est détectée car le récepteur sait quel bit, après le bit de départ, doit être un bit de stop 1 logique. Dans le cas où le bit de départ est réellement un bit de données, et que le bit de stop normal n'est pas un bit de stop mais un bit de départ ou de données, une erreur de perte d'encadrement de trame est signalée.

Erreurs de temporisation. Ce type de contrôle d'erreurs est exercé par le protocole de communication. Les temporisations sont les garants d'une communication synchrone entre les équipements. Lorsqu'un équipement source établit une communication, l'équipement destinataire doit répondre dans un certain laps de temps sans quoi, un dépassement de temps se produit qui annule la communication. Lors d'une communication avec le module CMM, il existe un certain nombre de cas dans lesquels un dépassement de temps se produira. Pour plus d'informations sur ces cas, voir Chapitre 7, Définition des protocoles.

2.3. TRANSMISSION ASYNCHRONE

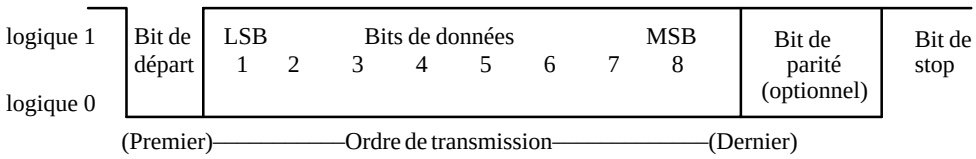
La transmission série asynchrone est utilisée dans les communications du module CMM. Bien qu'aucune horloge de synchronisation ne soit utilisée, les équipements émetteur et récepteur doivent opérer à la même vitesse sans quoi les erreurs mentionnées dans le paragraphe précédent se produiront.

Le format général des communications asynchrones comprend un bit de départ, huit bits de données, un bit de parité optionnel et un bit de stop.

Tableau 2-2. Format des données série

Format des données série										
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Bit 9	Bit 10
DEPART	LSB <———			BITS DE DONNEES ACTIFS				MSB ———>	PARITE (optionnelle)	STOP
0	<———			1	ou	0	———	———>		1

Lorsque le récepteur détecte le front avant du bit de départ, qui est toujours un 0 logique, un temporisateur est déclenché pour que l'échantillonnage se produise toujours au milieu de chaque bit. Une fois le dernier bit de données (ou le bit de parité) reçu, l'état logique de la liaison doit être à 1 pendant au moins la durée d'un bit avant réception du caractère qui suit. S'il n'y a plus de caractères à envoyer, le liaison reste à l'état 1.



2.4. LIAISON SÉRIE

La liaison série est le support physique par lequel les messages transiteront. Elle peut être fournie par une connexion physique directe entre les équipements ou une connexion par modems pour les communications longue distance. Les caractéristiques de la liaison dépendent des besoins de l'utilisateur et de la norme d'interface électrique en fonction de laquelle la liaison a été conçue.

2.4.1. Modems

Le mot **modem** est un acronyme pour MOdulateur/DEModulateur. Un modem est un équipement qui convertit les données, du numérique à l'analogique pour l'émission, et de l'analogique au numérique pour la réception, par les lignes téléphoniques. Certains modems utilisent d'autres méthodes de transmission, telles que la liaison radioélectrique ou les faisceaux hertziens.

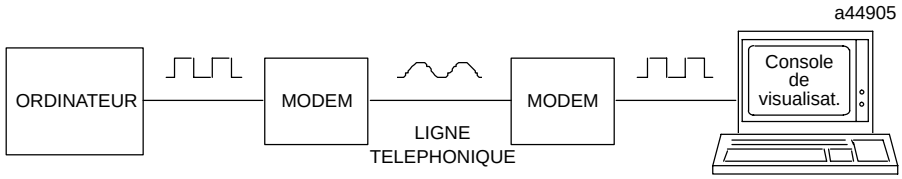


Figure 2-2. Modems utilisés dans la liaison

Les modems sont généralement classés suivant le type de lignes téléphoniques auxquelles ils peuvent être raccordés : half–duplex (bidirectionnel à l'alternat) ou duplex intégral (bidirectionnel simultané), synchrone ou asynchrone, technique de modulation utilisée pour le signal analogique et débit maximal en bits par seconde. A l'origine, les modems ont été conçus et le plus souvent utilisés avec l'interface RS–232D.

2.4.1.1. Modes de communication

Il existe trois modes de communication :

- Simplex : mode dans lequel les données ne peuvent être transmises que par une liaison unidirectionnelle.
- Half–duplex : mode dans lequel les données peuvent être transmises dans les deux sens par une liaison, mais dans un seul sens à la fois.
- Duplex intégral : mode dans lequel les données peuvent être transmises par une liaison dans les deux sens à la fois.

2.4.2. Interfaces normalisées

Une interface normalisée est un ensemble de règles qui définit les caractéristiques du signal, les caractéristiques du câble et de la connexion, les diverses fonctions affectées aux broches du connecteur, et les séquences de commande pour la liaison physique entre deux équipements. Les communications du module CMM se basent sur les interfaces normalisées décrites ci–dessous.

2.4.2.1. RS–232

Cette norme a été développée pour l'interconnexion d'un équipement terminal de traitement de données (ETTD), tel qu'une imprimante, une console de visualisation ou un ordinateur, et d'un équipement de terminaison de circuit de données (ETCD), tel qu'un modem, pour les transmissions par lignes téléphoniques ou par le réseau. Elle peut toutefois être utilisée pour des liaisons courte distance, sans modem. Du point de vue électrique, la norme RS–232 peut être décrite comme une interface de tension non équilibrée ou asymétrique. Cela signifie que tous les signaux de liaison partagent une terre électrique commune. Les principales caractéristiques de la norme RS–232 sont les suivantes :

- Longueur maximale du câble : 15 mètres
- Vitesse maximale : 20 Kilobits/secondes
- Affectations logiques référencées à la terre de signalisation :
  Espace ou logique 0 : +3 V à +25 V
  Marque ou logique 1 : –3 V à –25 V
- Connecteur 25 broches type D
- 21 circuits de liaison dont Emission et Réception de données, Commande de données, et Synchronisation. Les circuits les plus utilisés sont les suivants :

Tableau 2-3. Signaux de l'interface normalisée (RS–232)

Nº broche	Fonction	Abréviation	Type	Sens
1	Terre de protection	PROT GND	–	–
2	Emission de données	TXD	Données	De l'ETTD
3	Réception de données	RXD	Données	Vers ETTD
4	Demande pour émettre	RTS	Commande	De l'ETTD
5	Prêt à émettre	CTS	Commande	Vers ETTD
6	Modem prêt	DSR	Commande	Vers ETTD
7	Terre de signalisation	GND	–	–
8	Réception du détecteur de signal de ligne ou Détection de transmission de données	RLSD ou DCD	Commande	Vers ETTD
20	Terminal de données prêt	DTR	Commande	De l'ETTD

L'interface RS–232 peut être utilisée pour les liaisons directes n'excédant pas 15 mètres. Le schéma ci–après montre les liaisons requises pour que les deux équipements puissent émettre et recevoir.

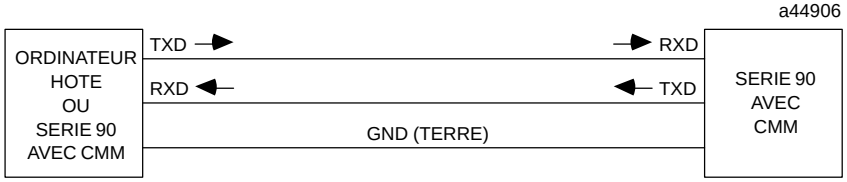


Figure 2-3. Liaison RS–232 directe sans contrôle de flux

Dans le schéma précédent, le flux de données n'est pas contrôlé ; en d'autres termes, les deux équipements peuvent émettre à tout instant et la disponibilité de la liaison n'est pas contrôlée avant émission.

Lorsque des modems sont utilisés, sans contrôle du flux des données, les deux équipements peuvent aussi émettre à tout instant, sans que la disponibilité de la liaison ou la présence de la porteuse ne soit contrôlée.

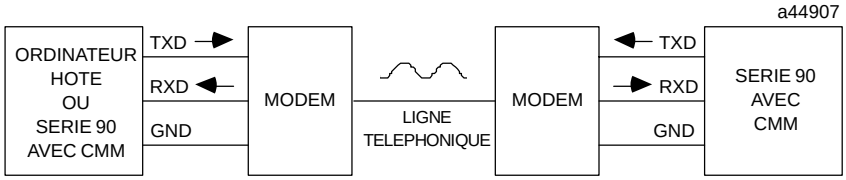


Figure 2-4. Liaison RS–232 avec modems sans contrôle de flux

Lorsque le contrôle de flux s'avère nécessaire, les commandes RTS et CTS peuvent être utilisées comme suit :

- RTS : L'équipement qui émet peut envoyer au modem une Demande pour émettre (des données).
- CTS : Le modem qui émet peut répondre à l'équipement qu'il est Prêt à émettre (les données).

Pour plus d'informations sur l'interconnexion de modules CMM à l'aide de modems, voir le paragraphe 5. Pour tout savoir sur l'utilisation des signaux de commande avec des modems et sur les caractéristiques mécaniques de l'interface, voir les Normes de l'interface électrique (EIA) RS–232D ainsi que le manuel d'utilisation fourni avec le modem.

2.4.2.2. RS–449, RS–422 et RS–485

RS–449, RS–422 et RS–485 désignent "une gamme de normes" qui reflètent les progrès accomplis en matière de technologie de circuits intégrés. Ces nouvelles normes permettent d'augmenter la distance entre les équipements et d'utiliser une vitesse maximale plus élevée. Les normes RS–422 et RS–485 définissent les caractéristiques de l'interface électrique. La norme RS–449, utilisée conjointement avec les normes RS–422 et RS–485, définit les diverses fonctions affectées aux broches du connecteur, les caractéristiques du câble et du connecteur ainsi que le séquencement des signaux de commande. RS–422 et RS–485 offrent des interfaces de tension équilibrées ou différentielles dans lesquelles les lignes de signalisation sont isolées de la terre. L'une des options d'interfaçage qui peut être utilisée dans la transmission série d'un automate Série 90 se base sur les normes RS–422, RS–485 et RS–449. Les principales caractéristiques des normes RS–422, RS–485 et RS–449 (référencées sous RS–485 dans ce manuel) sont les suivantes :

- Longueur maximale du câble : 1200 mètres.
- Vitesse maximale : 1 Mégabits/seconde à 1200 mètres et 10 Mégabits/seconde à 12 mètres.
- Affectations logiques ; entrées différentielles non référencées à la terre :

  Espace ou logique 0 : Le circuit A est de +200 mV à + 6 V par rapport au circuit B.

  Marque ou logique 1 : Le circuit A est de –200 mV à – 6 V par rapport au circuit B.
- Connecteur 25 broches type D.
- 30 circuits de liaison.

Pour plus d'informations sur les caractéristiques électriques et mécaniques de ces interfaces, voir les Normes EIA RS–449, RS–422 et RS–485.

3. LE PROTOCOLE DE COMMUNICATION

Comme expliqué au paragraphe 1, le protocole de communication est chargé d'établir et de libérer les connexions de messages et de traiter les messages suivant un ensemble de règles prédéfinies. Le protocole de communication est également chargé du contrôle d'erreurs au niveau des messages.

Ce paragraphe traite des sujets suivants :

- Protocoles de communication (SNP, SNP–X, CCM, RTU)
- Contrôle d'erreurs dans les messages : BCC, LRC, CRC

3.1. PROTOCOLES DE COMMUNICATION (SNP, SNP–X, CCM, RTU)

Les modules CMM Série 90–70 et Série 90–30 proposent quatre protocoles de communication : SNP, SNP–X, CCM, et RTU. Le protocole SNP a été développé pour la gamme d'API Série 90. Le protocole SNP gère les communications de type maître et esclave. Les modules CMM acceptent également le protocole SNP–X, extension optimisée du protocole SNP pour des transferts de données plus rapides.

Le protocole CCM (Communication Control Module) gère les communications de type égal à égal, maître et esclave.

Le protocole RTU (Modbus Remote Terminal Unit) est un protocole maître/esclave. Il permet de relier un API à un contrôleur de procédé, un ordinateur ou tout autre équipement intelligent opérant sous protocole RTU. En mode RTU, seul le maître peut déclencher une demande de communication. Le module CMM ne peut être configuré que comme un équipement RTU esclave.

3.2. PROTOCOLES SNP ET SNP–X

Les protocoles SNP et SNP–X sont des protocoles maître/esclave. Le maître déclenche la demande de communication et l'esclave y répond. Dans le contexte maître/esclave, le concept d'interconnexion logique est préservé.

Avant que des données puissent être transférées entre un équipement maître et un équipement esclave, le maître doit échanger un message de connexion avec l'esclave désigné. L'esclave désigné est identifié par l'identificateur ID SNP contenu dans la demande de connexion. Une fois que l'esclave désigné a renvoyé une réponse de connexion, l'interconnexion logique est terminée et le transfert de données peut commencer.

Cette interconnexion logique est obligatoire tant dans les configurations point–à–point que dans les configurations multipoint. Dans une configuration multipoint, chaque fois qu'un nouvel équipement esclave est sélectionné, une nouvelle interconnexion logique doit être établie. Avec le protocole SNP, ce processus peut être optimisé dans certains cas par un service de mise à jour du datagramme en temps réel. Ce service est décrit dans le Chapitre 6, Service SNP.

Une autre méthode d'optimisation de ce processus est fournie par les commandes SNP–X grâce auxquelles des interconnexions logiques simultanées avec plusieurs équipements esclaves peuvent être maintenues. Ce service est décrit dans le Chapitre 6, Service SNP–X.

Le processus d'interconnexion peut être simplifié dans les configurations point–à–point en spécifiant un identificateur ID SNP nul dans la demande de connexion de l'équipement maître. Tous les équipements SNP/SNP–X esclaves doivent répondre à un ID SNP nul, quel que soit l'ID SNP qui lui a été affecté. Ainsi, un équipement maître peut s'interconnecter à n'importe quel équipement esclave sans connaître son ID SNP. Les ID SNP nuls ne peuvent toutefois être utilisés *que dans* des configurations point–à–point.

Pour plus d'informations sur le protocole SNP, avec schémas de transfert de messages, voir Chapitre 7, § 3, Protocole SNP.

Avec le protocole SNP–X, certaines améliorations ont été apportées au protocole SNP qui simplifient l'utilisation et améliorent la performance. Les accès en lecture/écriture typiques aux tables de références automate sont plus simples et plus rapides. Plus simples parce que SNP–X fournit une interface de programmation COMREQ moins complexe et unique. Plus rapides parce que SNP–X réduit le temps système superflu du protocole et supporte les interconnexions logiques

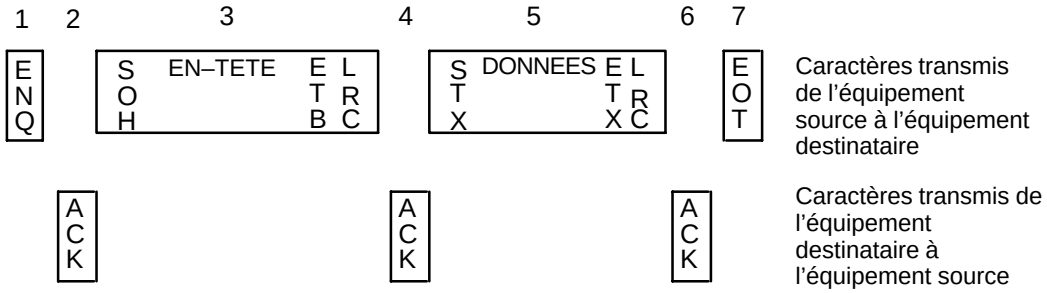
simultanées avec plusieurs équipements SNP–X esclaves. A la différence du protocole SNP sous licence, le protocole SNP–X est décrit en détail dans le Chapitre 7, § 4, Protocole SNP–X, qui fournit en outre toutes les informations nécessaires pour la mise en oeuvre d'un gestionnaire de communication SNP–X maître.

Code de contrôle de bloc BCC. Chaque message SNP/SNP–X renferme un octet BCC qui permet de vérifier que le message a été correctement reçu. Une erreur BCC dans un message SNP reçu génère un accusé de réception négatif.

3.3. PROTOCOLE CCM

Le schéma ci–dessous montre comment un message est formaté sous le protocole CCM en mode Egal–à–égal. Pour plus d'informations sur le protocole CCM, voir Chapitre 7, § 1.

Le message ci–après demande un transfert de données de l'équipement source (initiateur) à l'équipement destinataire (répondeur) :



1. ENQ(uire) est un caractère de commande ASCII qui signifie Demande de renseignement et cherche à déterminer si l'équipement destinataire est prêt.
2. ACK(nowledge) est un caractère de commande ASCII qui signifie Accusé de réception positif. (L'équipement est prêt à communiquer).
3. Le bloc d'en–tête renferme les informations suivantes codées en ASCII :
   - SOH : Caractère de commande ASCII qui signifie Start of Header (début d'en–tête).
   - ID de l'équipement destinataire.
   - Sens du transfert de données.
   - Type de données transférées.
   - Adresse mémoire destinataire pour les données transférées.
   - Volume de données transférées.
   - ID de l'équipement source.
   - ETB : Caractère de commande ASCII qui signifie End of Transmission Block (fin de bloc).
   - LRC : Contrôle de redondance longitudinale.
4. ACK : Accusé de réception positif, les informations contenues dans l'en–tête sont valides.
5. Le bloc de données renferme les informations suivantes :
   - STX : Caractère de commande ASCII qui signifie Start of Text (début de texte).
   - Données binaires non codées.
   - ETX : Caractère de commande ASCII qui signifie End of Text (fin de texte).
   - LRC : Contrôle de redondance longitudinale.
6. ACK : Accusé de réception positif, les informations contenues dans les données sont valides.
7. EOT : Caractère de commande ASCII qui signifie End of Transmission (fin de transmission).

Contrôle d'erreur par redondance longitudinale. Le contrôle de redondance longitudinale (LRC) est une méthode de détection d'erreurs sur des blocs complets. Le protocole CCM utilise cette méthode pour contrôler les erreurs sur des blocs de données. L'équipement qui émet calcule le LRC et l'insère à la fin du bloc d'en–tête et de chaque bloc de données. L'équipement qui reçoit génère son propre LRC d'après les données en entrée et le compare au LRC transmis. S'ils diffèrent, c'est qu'une erreur s'est produite au cours de la transmission.

Le LRC est généré en appliquant une porte OU exclusif logique (XOR) sur chaque octet du bloc d'en–tête ou de données. Le protocole CCM ne tient pas compte des octets SOH et ETB du bloc d'en–tête, ni des octets STX et ETB/ETX du bloc de données dans le calcul du LRC. L'exemple ci–après montre comment le LRC est déduit pour un bloc de données comportant trois octets de données :

Bit de parité	Octet de données								
(impaire)	8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	1	– 1er caractère de données transmis
0	0	0	0	0	0	0	1	0	– 2ème caractère de données transmis
	0	0	0	0	0	0	1	1	– Résultat XOR des 1er et 2ème caractères de données
1	0	0	0	0	0	1	0	1	– 3ème caractère de données transmis
	0	0	0	0	0	1	1	0	– LRC=XOR du précédent XOR et du 3ème caractère de données

3.4. PROTOCOLE RTU

Le protocole RTU est un protocole de type interrogation–réponse. Le schéma ci–après montre comment un message est formaté sous le protocole RTU :

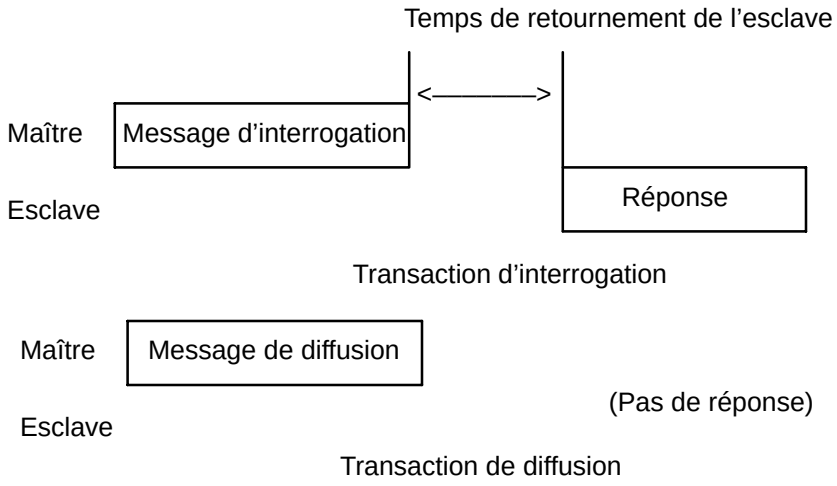
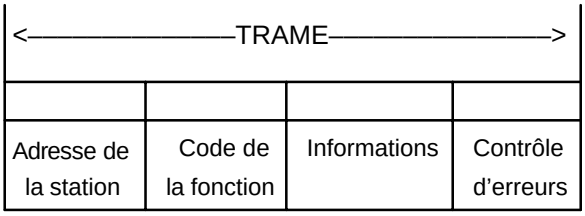


Figure 2-5. Transferts de messages RTU

Une distinction est établie entre les équipements qui communiquent. L'équipement qui déclenche le transfert de données est appelé l'équipement maître et l'autre, l'équipement esclave. Le module CMM ne peut être configuré que comme un équipement RTU esclave.

Avant tout transfert de données, l'équipement maître envoie un message d'interrogation ou de demande de diffusion. L'équipement esclave accepte le transfert de données en envoyant un message Réponse si l'équipement maître lui a adressé un message d'interrogation, et en ne répondant pas si l'équipement maître lui a adressé une demande de diffusion. Le laps de temps qui s'écoule entre la fin d'une interrogation et le début de la réponse à cette interrogation est appelé Temps de retournement de l'esclave.

Les champs typiques d'un message RTU sont illustrés ci–dessous :



Contrôle de redondance cyclique. Le champ Contrôle d'erreurs d'un message RTU renferme un code de contrôle de redondance cyclique (CRC–16). Pour plus d'informations sur la génération de ce code, voir Chapitre 7, § 2, Protocole RTU.

4. L'INTERFACE D'APPLICATION API

Comme expliqué au paragraphe 1, l'interface d'application API fait le lien entre le protocole de communication et le programme utilisateur dans l'API. C'est par l'interface d'application API que le programme utilisateur établit l'échange de communication. Le mécanisme d'acheminement de l'échange est fourni par la fonction COMREQ (COMmunications REQuest – Demande de communication).

La fonction COMREQ permet au programme utilisateur de passer des paramètres spécifiant la demande au protocole de communication. Dans certains cas, la fonction COMREQ comporte des paramètres qui spécifient l'adresse de la table de références pour la réponse qui suit. Dans tous les cas, la fonction COMREQ spécifie également des renseignements supplémentaires indispensables à la demande, tels que l'adresse du mot d'état dans lequel sera stocké le résultat de chaque transaction.

Les fonctions COMREQ utilisées par les modules CMM sont décrites dans le Chapitre 4.

Outre le traitement de la fonction COMREQ, l'interface d'application API maintient le mot d'état et accède aux diverses tables de références, selon les besoins. Ces opérations se font en dehors du programme utilisateur et fournissent ainsi aux équipements esclaves les mécanismes leur permettant de communiquer avec l'API. Elles se chargent des transferts matériels sur le bus, des demandes de Boîtes aux lettres VME (Mailbox) et des services de l'API Série 90–70, ainsi que des demandes émanant du bus du module intelligent de l'API Série 90–30.

5. LES RÉSEAUX DE COMMUNICATION

L'expression "Configuration du réseau" désigne la façon dont les ordinateurs, les terminaux et les équipements de communication sont interconnectés. Avec les API Série 90, les configurations de systèmes de communication des données suivantes sont possibles :

- Point–à–Point (mode égal–à–égal ou maître–esclave)
- Multipoint (Un seul maître–plusieurs esclaves)
- Transmission par modem

5.1. POINT–À–POINT

La liaison point–à–point est la configuration système la plus simple ; avec cette méthode, seuls deux équipements sont connectés sur la même liaison. Une interface normalisée RS–232 ou RS–422 peut être utilisée. Les deux équipements peuvent être connectés en mode égal–à–égal, auquel cas les deux peuvent établir la communication, ou en mode maître–esclave où seul l'équipement maître peut établir la communication. Les schémas ci–dessous illustrent le principe de la configuration point–à–point :

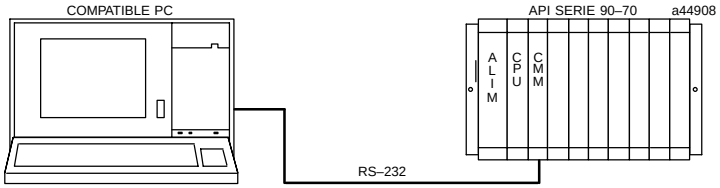


Figure 2-6. Exemple de configuration point–à–point RS–232

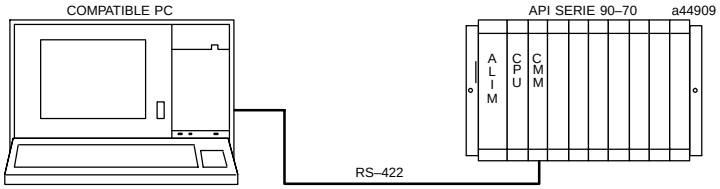


Figure 2-7. Exemple de configuration point–à–point RS–422/485

5.2. MULTIPOINT

La configuration multipoint offre une structure de ”ligne collective” dans laquelle plusieurs équipements partagent la même liaison. Dans un réseau multipoint câblé, tous les équipements doivent comporter une interface RS–422. Si des convertisseurs ou des modems sont utilisés pour connecter les équipements au réseau, l’interface RS–232 ou RS–422 peut être utilisée. Il est important de se souvenir que l’interface RS–232 ne peut être utilisée qu’entre deux équipements ; lorsqu’un équipement est physiquement raccordé à plus d’un équipement, l’interface RS–422 doit être utilisée. L’un des équipements est l’équipement maître et les autres sont les équipements esclaves ; seul l’équipement maître peut établir la communication avec les autres équipements du système. Le schéma ci–dessous illustre le principe de la configuration multipoint :

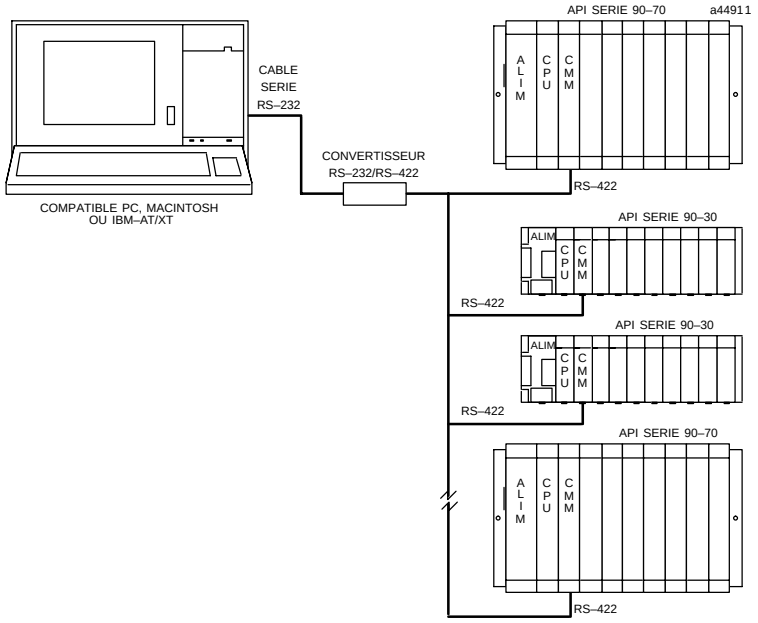


Figure 2-8. Exemple de configuration système multipoint

Dans une configuration multipoint, sous protocole SNP ou CCM, l’un des modules CMM ou l’équipement–hôte est configuré comme l’équipement maître et un ou plusieurs modules CMM sont configurés comme des équipements esclaves ; seul le mode maître–esclave est supporté. Le module CMM configuré comme l’équipement maître peut établir les communications ; les équipements esclaves ne le peuvent pas. Sous protocole RTU, l’équipement–hôte capable d’émuler le protocole RTU est l’équipement maître et le ou les modules CMM en mode RTU sont les équipements esclaves.

Les équipements esclaves inactifs surveillent continuellement la liaison pour déterminer si elle est occupée ou libre. Sous le protocole SNP, lorsque la liaison est libre, les esclaves commencent par rechercher une séquence Break. Dès qu’un Break est reçu, chaque esclave recherche alors la demande de connexion qui doit suivre. Seul l’esclave dont l’ID SNP correspond à celui figurant dans la demande d’interconnexion répondra. Sous le protocole CCM, lorsque la liaison est libre, les esclaves commencent par rechercher les nouvelles séquences de demande de renseignement (ENQ). Comme généralement plusieurs équipements esclaves partagent la même liaison multipoint, chaque esclave ne reconnaîtra que les demandes de renseignement contenant son propre identificateur. Sous le protocole RTU, les esclaves rechercheront les nouvelles demandes. Comme généralement plusieurs équipements esclaves partagent la même liaison multipoint, chaque esclave ne traitera que les demandes contenant sa propre adresse de station, ou l’adresse de diffusion destinée à tous les équipements esclaves.

5.3. TRANSMISSION PAR MODEM

Le mot **modem** est un acronyme pour MOdulateur/DEModulateur. Un modem est un équipement qui convertit les données du numérique à l'analogique pour l'émission et de l'analogique au numérique pour la réception par les lignes téléphoniques.

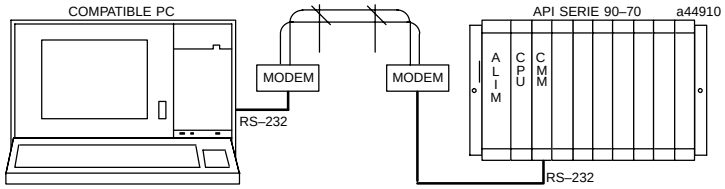


Figure 2-9. Exemple de configuration système avec transmission par modem

Certains modems utilisent d'autres méthodes de transmission telles que les liaisons radioélectriques ou les faisceaux hertziens. Ces types de modems sont principalement utilisés lorsque les modems ne peuvent pas être raccordés par des câbles. Certains pays ont des exigences particulières. Par exemple, en règle générale, la Commission fédérale de communications américaine (FCC) exige l'utilisation de transmetteurs monofréquence avec des temps de transmission courts. C'est pourquoi un temps de démarrage doit être ajouté avant chaque transmission pour les transmetteurs radio. Le module CMM accorde les transmetteurs radio pour qu'ils terminent l'état transitoire et attendent un peu avant de commencer à transmettre les données. Les diverses temporisations du protocole de communication sont augmentées pour tenir compte de ce délai supplémentaire.

Avec les transmetteurs qui utilisent les faisceaux hertziens ou les liaisons radioélectriques, le câblage varie suivant le type de modems et de transmetteurs utilisés. Pour plus d'informations à ce sujet, renseignez–vous auprès de votre vendeur ou de votre agence après–vente GE Fanuc Automation.

Page laissée blanche intentionnellement

Chapitre 3 Module CMM – Description, Installation et Configuration

Ce chapitre décrit les modules coprocesseurs de communication (CMM311, CMM711) et explique comment les installer et les configurer. Il explique également comment fonctionnent les modules, une fois installés dans des API Série 90 :

Ce chapitre traite des sujets suivants :

- 1: Description matérielle du module CMM et fonctionnement
- 2: Installation et Configuration du module CMM

1. DESCRIPTION MATÉRIELLE DU MODULE CMM ET FONCTIONNEMENT

Ce paragraphe qui s'applique à la fois au module CMM711 pour l'API Série 90–70 et au module CMM311 pour l'API Série 90–30 traite des sujets suivants :

- Voyants
- Bouton de Restart/Reset
- Connecteurs série
- Fonctionnement du système

Les schémas ci–dessous montrent comment se présentent les modules CMM :

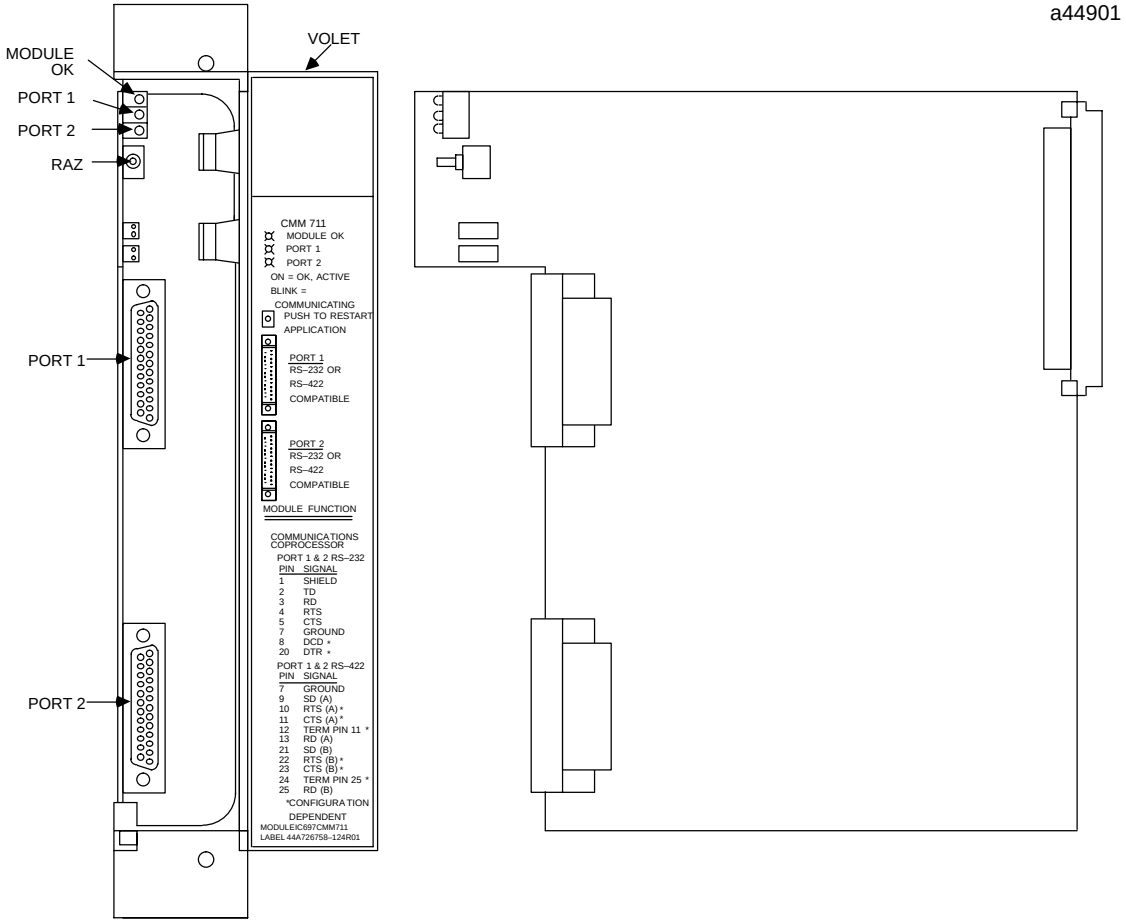


Figure 3-1. Module CMM Série 90–70 (CMM711)

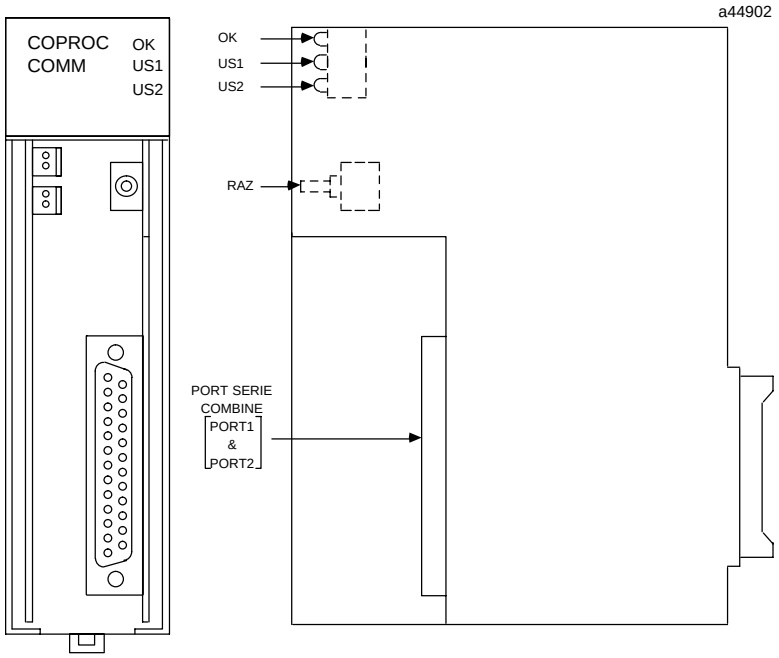


Figure 3-2. Module CMM Série 90–30 (CMM311)

A part les connecteurs de port série, les interfaces utilisateur des modules CMM311 et CMM711 sont identiques. Le module CMM711 Série 90–70 est équipé de deux connecteurs de port série. Le module CMM 311 Série 90–30 n'a qu'un seul connecteur de port série qui fournit deux ports. Chaque interface utilisateur est décrite en détail ci–dessous.

1.1. VOYANTS

Les trois voyants, illustrés dans les schémas ci–dessus, se situent sur le bord avant supérieur de la carte CMM.

1.1.1. Voyant Module OK

Le voyant MODULE OK indique l'état courant de la carte CMM. Il peut prendre trois états :

Eteint : Lorsque le voyant est éteint, le module CMM ne fonctionne pas. Cela indique un problème matériel (exemple : les tests de diagnostic ont détecté un incident, le module CMM est en panne, ou l'API n'est pas présent). Une action correctrice s'impose pour remettre le module CMM en état de marche.

Allumé : Lorsque le voyant est allumé en continu, le module CMM fonctionne correctement. En règle générale, ce voyant doit toujours être allumé ; il indique ce faisant que les tests de diagnostic n'ont détecté aucun incident et que les données de configuration du module sont correctes.

Clignotant : Le voyant clignote pendant que les diagnostics de mise sous tension s'exécutent.

1.1.2. Voyants Ports série

Les deux autres voyants, PORT1 et PORT2 (US1 et US2 pour le module CMM311 Série 90–30), clignotent lorsque les ports série sont actifs. PORT1 (US1) clignote lorsque le port 1 envoie ou reçoit des données ; PORT2 (US2) clignote lorsque le port 2 envoie ou reçoit des données.

1.2. BOUTON DE RESTART/RESET

Si vous appuyez sur le bouton de Restart/Reset alors que le voyant MODULE OK est allumé, le module CMM sera réinitialisé à ses valeurs de configuration logicielle par défaut.

Si le voyant MODULE OK est éteint (incident matériel), le bouton de Restart/Reset n'a aucun effet ; l'API doit être redémarré pour que la carte CMM se remette à fonctionner.

1.3. PORTS SÉRIE

Les ports série du module CMM permettent de communiquer avec des équipements externes. Le module CMM (CMM711) Série 90–70 est équipé de deux ports série, avec un connecteur pour chaque port. Le module CMM (CMM311) Série 90–30 est également équipé de deux ports série, mais n'a qu'un seul connecteur. Les ports série et les connecteurs pour chaque API sont décrits ci–dessous.

1.3.1. Ports série du module CMM711

Le module CMM Série 90–70 est équipé de deux ports série qui supportent à la fois les modes RS–232 et RS–485. Les fonctions affectées aux broches des deux ports sont identiques.

Remarque

En mode RS–485, le module CMM peut être connecté à des équipements RS–422 ainsi qu'à des équipements RS–485.

Les fonctions affectées aux broches des connecteurs des deux ports du module CMM Série 90–70 sont décrites ci–dessous :

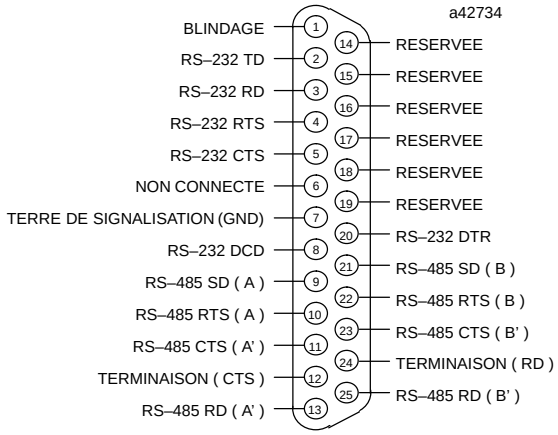


Figure 3-3. Fonctions affectées aux broches des ports série du module CMM Série 90–70

Remarque

Dans le schéma ci–dessus, (A) et (B) sont identiques pour – et +. A' et B' représentent les entrées, et A et B, les sorties. Pour terminer le signal d'entrée RS–485 CTS, reliez les broches 11 et 12 ; pour terminer le signal d'entrée RD, reliez les broches 24 et 25.

1.3.2. Ports série du module CMM311

Le module CMM Série 90–30 est équipé d'un seul connecteur série qui supporte deux ports. Les applications du Port 1 doivent utiliser l'interface RS–232. Les applications du Port 2 peuvent utiliser l'interface RS–232 ou RS–485.

Remarque

En mode RS–485, le module CMM peut être connecté à des équipements RS–422 ainsi qu'à des équipements RS–485.

Les fonctions affectées aux broches du connecteur du module CMM Série 90–30 sont décrites ci–dessous :

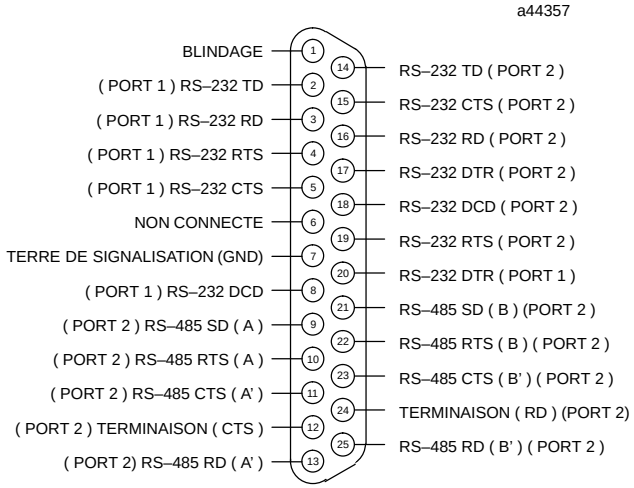
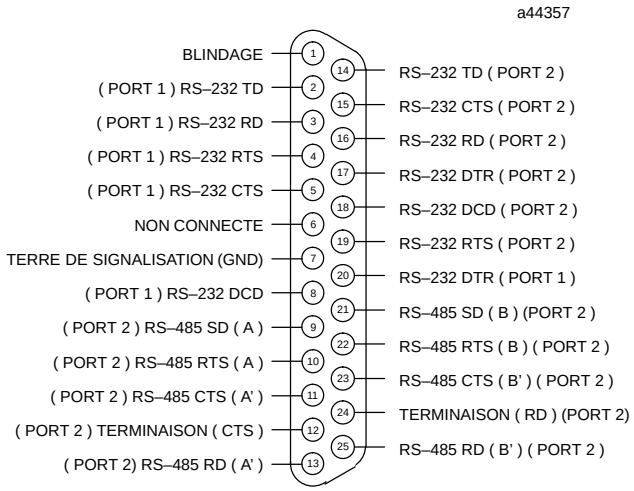


Figure 3-4. Fonctions affectées aux broches des ports série du module CMM Série 90–30

Les signaux RS–485 du port 2 et les signaux RS–232 du port 1 sont affectés aux broches standard du connecteur. Les signaux RS–232 du port 2 sont affectés aux broches normalement inutilisées du connecteur.

Un câble en "Y" est fourni avec chaque module CMM Série 90–30. Il sert à séparer les deux ports fournis sur un seul connecteur physique (plus précisément, à séparer les signaux). De plus, ce câble en "Y" rend les câbles utilisés avec le module CMM Série 90–70 entièrement compatibles avec le module CMM Série 90–30.

Ce câble en "Y" de 30 cm est équipé d'un connecteur coudé à angle droit du côté du port série du module CMM. L'autre extrémité présente un connecteur double ; un pour le port 1, l'autre pour le port 2.



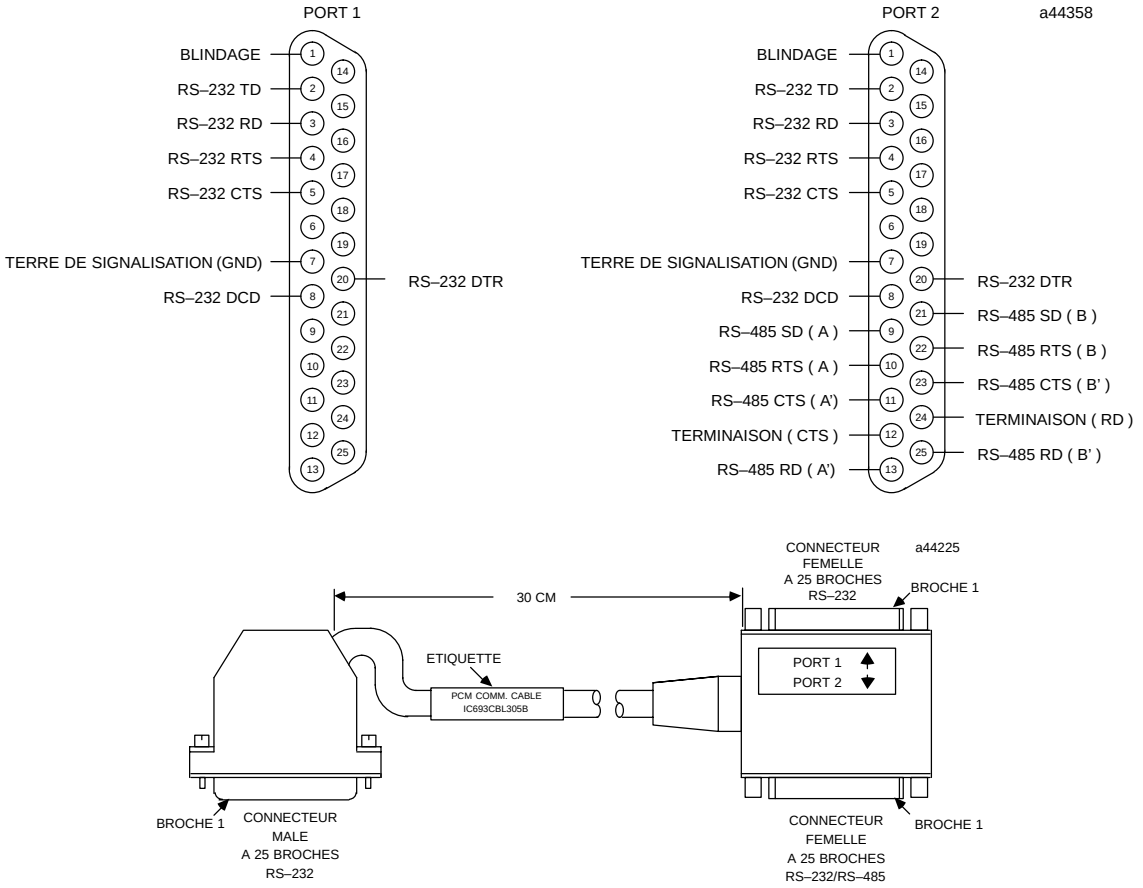


Figure 3-5. Branchements du câble en ”Y” du module CMM Série 90–30

Le câble en ”Y” achemine les signaux RS–232 des Ports 1 et 2 sur les broches RS–232 désignées. Si vous n’utilisez pas ce câble, il vous faudra fabriquer un câble spécial pour raccorder les équipements RS–232 au Port 2. Les câbles standard du module CMM Série 90–70 peuvent être utilisés pour le module CMM Série 90–30 si le câble en ”Y” est conservé.

1.4. FONCTIONNEMENT SYSTÈME DU MODULE CMM

Ce paragraphe explique le fonctionnement du module CMM lorsqu’il est installé dans un API Série 90. Il traite des sujets suivants :

- Fonctionnement système du module CMM Série 90–70
- Fonctionnement système du module CMM Série 90–30

1.4.1. Fonctionnement système du module CMM Série 90–70

Le module CMM Série 90–70 est équipé d’un microprocesseur 80186.

La mémoire du module CMM Série 90–70 est constituée de PROM, RAM locale, et RAM partagée. La mémoire PROM renferme le système d’exploitation (OS), les protocoles de communication et les utilitaires associés. L’usage de la mémoire RAM locale est réservé au module CMM. La mémoire RAM partagée sert aux communications entre le module CMM et l’API Série 90–70.

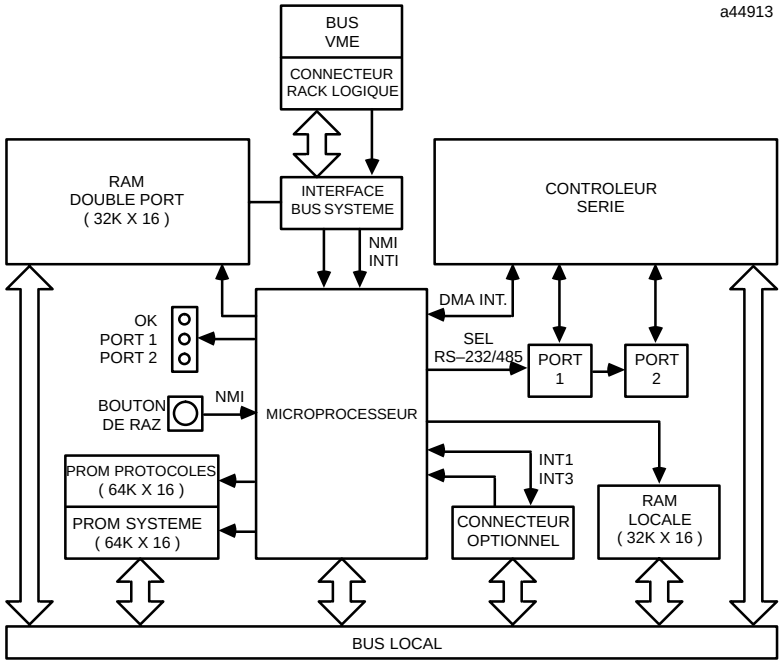


Figure 3-6. Schéma de principe du module CMM Série 90–70

1.4.2. Fonctionnement système du module CMM Série 90–30

Le microprocesseur 8 bits à haute intégration 80188 est au coeur de la carte CMM Série 90–30.

La mémoire du module CMM est constituée de PROM et de RAM locale. La mémoire PROM renferme le système d'exploitation (OS), les protocoles de communication et les utilitaires associés. L'usage de la mémoire RAM locale est réservé au module CMM.

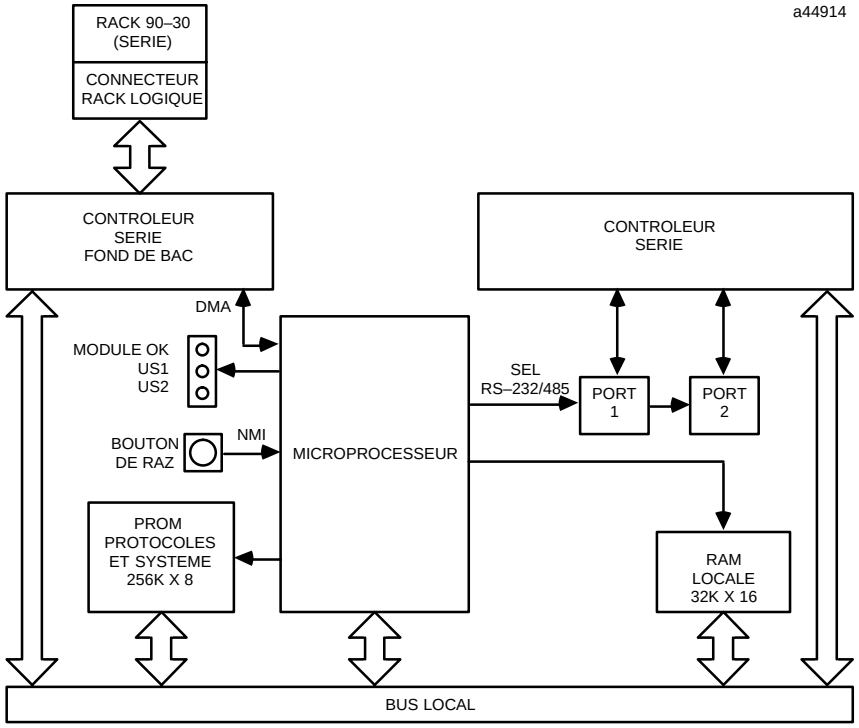


Figure 3-7. Schéma de principe du module CMM Série 90–30

2. INSTALLATION ET CONFIGURATION DU MODULE CMM

Ce paragraphe explique comment installer le module CMM dans le rack et configurer le module à l'aide du logiciel Logicmaster. Il traite des sujets suivants :

- Configuration requise
- Installation du module CMM
- Configuration du module CMM avec le logiciel Logicmaster 90
- Configuration du module CMM311 avec la miniconsole de programmation HHP.

2.1. CONFIGURATION REQUISE

Pour pouvoir installer un module CMM, la configuration matérielle et logicielle suivante est requise :

- Automate programmable industriel (API) Série 90–70 ou Série 90–30.
  (Pour un API Série 90–30, la CPU doit être de type CPU331 ou supérieur.)
- Module coprocesseur de communication (CMM) à installer et à tester :
  - IC697CMM711 pour un 90–70
  - IC693CMM311 pour un 90–30

Si l'application utilise le protocole CCM avec les paramètres standard définis par défaut pour le module CMM (CMM 311) Série 90–30, aucun autre équipement n'est requis.

Pour les autres applications, vous devez également disposer de :

- Un ordinateur avec disque dur. Vous avez le choix entre :
  - Un ordinateur personnel IBM PC/XT ou AT ou équivalent compatible doté d'un clavier de 83 ou 101 touches.
  - Un IBM AT industriel ou ordinateur compatible doté d'un clavier de 101 touches en standard.
- Un système Logicmaster 90 comprenant le logiciel, le matériel et les câbles appropriés. Pour plus d'informations à ce sujet, voir le document *GFK–0263 Logicmaster 90–70 Programming Software User's Manual*, ou le document *GFK–0466 Logicmaster™ 90–30/20/Micro Programming Software User's Manual*.

2.2. INSTALLATION DU MODULE CMM

La première étape consiste à installer physiquement la carte CMM et à vérifier qu'elle fonctionne correctement.

2.2.1. Généralités

Dans un système à un seul rack, le module CMM réside dans le même rack que la CPU. Dans un système Série 90–70 à plusieurs racks, le module CMM peut résider soit dans le même rack que la CPU, soit dans un rack d'extension. Le module CMM Série 90–30 doit résider dans le même rack que la CPU.

Le schéma ci–dessous montre des configurations système possibles dans lesquelles un module CMM Série 90–70 est installé dans un rack local ou d'extension :

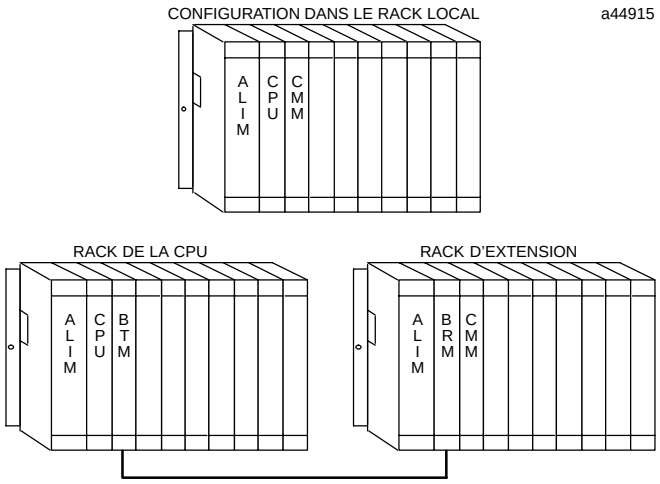


Figure 3-8. Configurations d'un module CMM Série 90–70

L'alimentation, la CPU et le module d'extension de bus BEM (Bus Expansion Module) Série 90–70 occupent des emplacements spécifiques dans le rack. Le terme Module d'extension de bus BEM désigne à la fois les modules émetteur sur bus BTM (Bus Transmitter Modules) et les modules récepteur sur bus BRM (Bus Receiver Modules). Le module CPU doit être installé dans l'emplacement 1 du rack 0.

Le système Série 90–70 inclut généralement un module émetteur sur bus BTM (Bus Transmitter Module). La version A du module émetteur sur bus BTM doit être installée à droite de toutes les autres cartes GE Fanuc ; la version B doit être installée dans l'emplacement 2 du rack 0. Si le système d'API comporte plus d'un rack, un module récepteur sur bus BRM (Bus Receiver Module) doit être installé dans l'emplacement 1 de chaque rack d'extension.

Le module CMM peut être installé dans n'importe quel emplacement non réservé d'un rack, pourvu que les conditions suivantes soient respectées :

- Les fichiers de configuration créés par le logiciel de configuration Logicmaster 90 doivent refléter la configuration physique des modules. Dans le cas contraire, l'automate risque de ne pas fonctionner comme il se doit. Tous les défauts seront enregistrés dans la table des défauts automate. Pour plus d'informations sur la création de fichiers de configuration avec le logiciel Logicmaster 90, voir les manuels GFK–0263 ou GFK–0466.
- Lorsque les modules CMM sont installés dans le rack d'un API Série 90–70, tous les emplacements entre le module CMM et la CPU de l'API doivent être occupés. Si certains d'entre eux sont vides, le module CMM ne pourra pas communiquer par le fond de bac avec la CPU de l'API Série 90–70.
- Le module CMM Série 90–30 doit être installé dans le rack principal avec la CPU de l'API.

2.2.2. Installation du module CMM dans le rack

Pour installer le module CMM dans le rack, procédez comme suit :

1. Pour un API Série 90–70, positionnez le commutateur Run/Stop de la CPU sur STOP. Pour un API Série 90–30, utilisez le logiciel Logicmaster 90 ou la miniconsole de programmation HHP pour arrêter l'API. Cela empêche l'éventuel programme d'application local de l'API de générer des commandes susceptibles d'affecter le fonctionnement du module.
2. Mettez le système d'API Série 90 hors tension.
3. Localisez le rack et l'emplacement désirés.
4. Glissez le module CMM dans l'emplacement choisi, de façon que les trois voyants soient en haut de la carte.
5. Bloquez la carte en exerçant une pression ferme, mais pas excessive, dessus.
6. Mettez le rack de l'API sous tension. Le voyant du haut (MODULE OK) du module Série 90–70 clignotera pendant que les diagnostics de mise sous tension s'exécutent. Il continuera à clignoter en attendant que la CPU lui fournisse les données de configuration. Si aucun signal n'est reçu par le fond de bac au bout de 30 secondes, le module CMM Série 90–70 supposera que la CPU n'est pas là et continuera sa séquence de mise sous tension sans lui. Une fois que le module CMM est prêt, ce voyant doit cesser de clignoter et rester ALLUME.

Remarque

Le voyant du haut (MODULE OK) du module CMM Série 90–30 ne s'allume pas si l'API n'est pas présent ou si le module CMM n'a pas subi ses diagnostics de mise sous tension avec succès.

7. Répétez la présente procédure pour chaque module CMM.

2.2.3. Dépannage

Une fois que vous avez installé le module CMM comme expliqué ci–dessus, le voyant MODULE OK doit rester allumé en continu. Si tel n'est pas le cas, c'est que la lampe du voyant est grillée ou que la carte n'a pas subi ses diagnostics de mise sous tension avec succès. Pour déterminer l'origine du problème, procédez comme suit :

1. Vérifiez que le rack est sous tension, que l'API fonctionne correctement et que le module CMM est correctement installé.
2. Vérifiez le contenu de la table des défauts automate à l'aide du logiciel Logicmaster 90. Si elle comporte une ligne "module défectueux ou absent", la carte peut être défectueuse ; sinon, appelez GE Fanuc pour obtenir de l'aide.

2.3. CONFIGURATION DU MODULE CMM AVEC LE LOGICIEL LOGICMASTER 90

La procédure de configuration du module CMM se décompose en deux étapes :

- Configuration du rack d'E/S
- Configuration du module (configuration logicielle)

Remarque

Si vous configurez un module CMM311 à l'aide d'une miniconsole de programmation HHP, sautez cette procédure. La procédure de configuration avec une miniconsole est décrite plus loin dans ce chapitre.

2.3.1. Configuration du rack d'E/S

Le logiciel de configuration du rack d'E/S Série 90 permet de définir le type et l'emplacement de tous les modules installés dans les racks de l'API. Pour ce faire, vous devez renseigner les écrans de configuration qui représentent les modules installés dans un rack. Vous pouvez sélectionner des paramètres de configuration spécifiques pour les modules individuels. Les données de configuration sont sauvegardées dans un fichier de configuration qui est ensuite stocké dans l'API Série 90.

2.3.1.1. API Série 90–70

Tout rack d'un API Série 90–70 peut être configuré comme rack de la CPU (rack principal) ou comme rack d'extension. Chaque rack comporte 5 ou 9 emplacements destinés aux modules, plus un emplacement (à l'extrême–gauche) pour l'alimentation ou le branchement d'une alimentation. Le rack illustré ci–après comporte 9 emplacements pour les modules.

Le rack où est installé la CPU doit comporter une alimentation dans l'emplacement Alim et un module CPU dans l'emplacement 1. L'emplacement 1 du rack de la CPU est réservé au module CPU ; dans un rack d'extension, l'emplacement 1 est réservé au module récepteur sur bus BRM (Bus Receiver Module). Un module émetteur sur bus BTM (Bus Transmitter Module) doit obligatoirement être installé dans l'emplacement 2 du rack 0 s'il s'agit d'une version B ; s'il s'agit d'une version A, le module BTM doit être installé à droite de toutes les autres cartes GE Fanuc. Le module émetteur sur bus BTM (Bus Transmitter Module) est requis pour les communications parallèles avec la console de programmation ou dans le cas d'un système à plusieurs racks.

Les emplacements restants peuvent contenir toute combinaison de modules d'E/S ou modules intelligents nécessaire au programme d'application. Le module CMM peut être installé dans n'importe quel emplacement, excepté l'emplacement 1, du rack de votre choix. Il ne doit pas y avoir d'emplacements vides à gauche des modules optionnels ou des modules d'E/S qui génèrent des interruptions dans un système d'API Série 90–70. Si le système nécessite plus de modules d'E/S qu'il n'y a d'emplacements dans un rack, des racks supplémentaires peuvent être ajoutés, à concurrence de huit maximum (y compris le rack de la CPU).

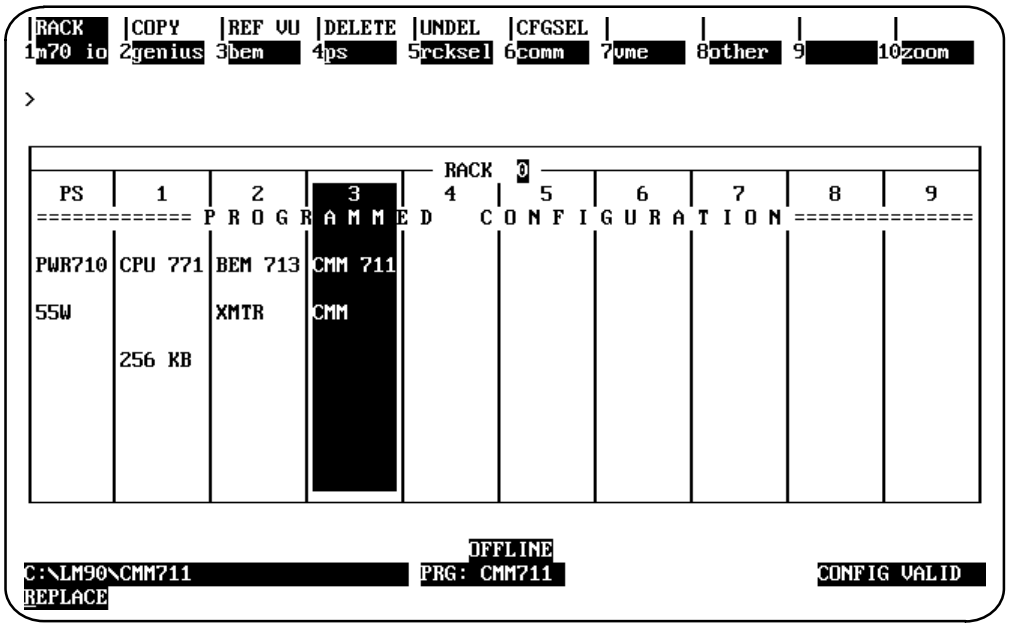
2.3.1.2. API Série 90–30

Pour les API Série 90–30, le module CMM doit être installé dans le rack de la CPU. L'emplacement le plus à gauche est réservé à l'alimentation et l'emplacement 1 au module CPU ; exception faite de ces deux emplacements, le module CMM peut être installé n'importe où. Il n'y a pas de module d'extension de bus BEM (Bus Expansion Module) comme dans les API Série 90–70.

2.3.1.3. Accès à l'écran de configuration du rack d'E/S

1. Commencez par sélectionner le logiciel de configuration dans le logiciel Logicmaster 90.
2. Une fois dans le menu de configuration principal, appuyez sur F1 (E/S).

L'écran de configuration du rack d'E/S montre les divers modules installés dans le système d'API Série 90. L'écran ci–dessous montre un système d'API Série 90–70 dans lequel un module CMM711 et divers autres modules sont déjà installés :



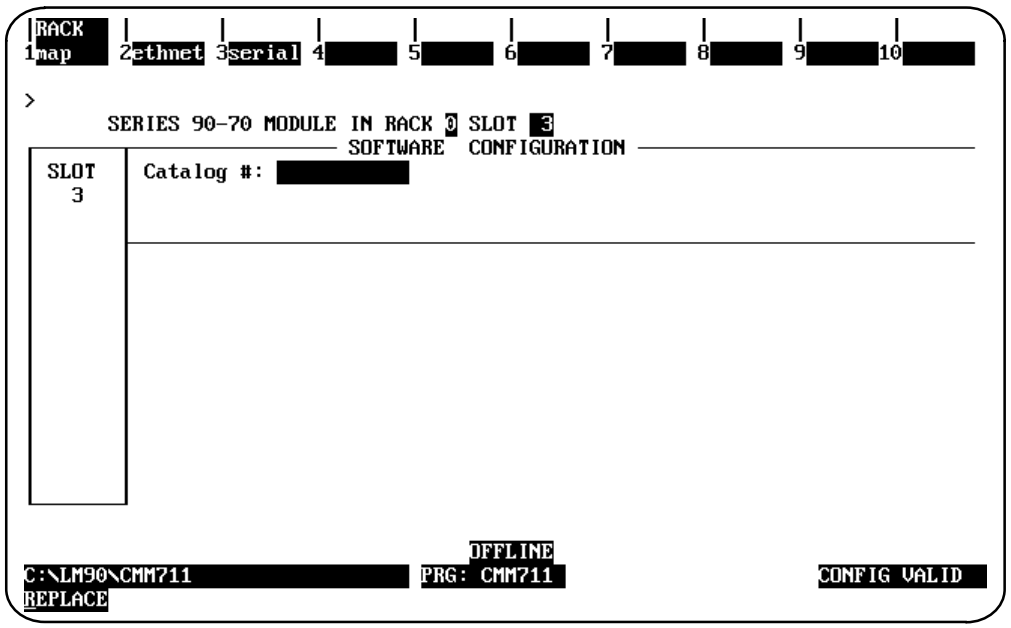
2.3.1.4. Fonctionnement de l'écran de configuration du rack d'E/S

Les principales opérations qui s'appliquent à l'écran de configuration du rack d'E/S sont décrites ci–dessous :

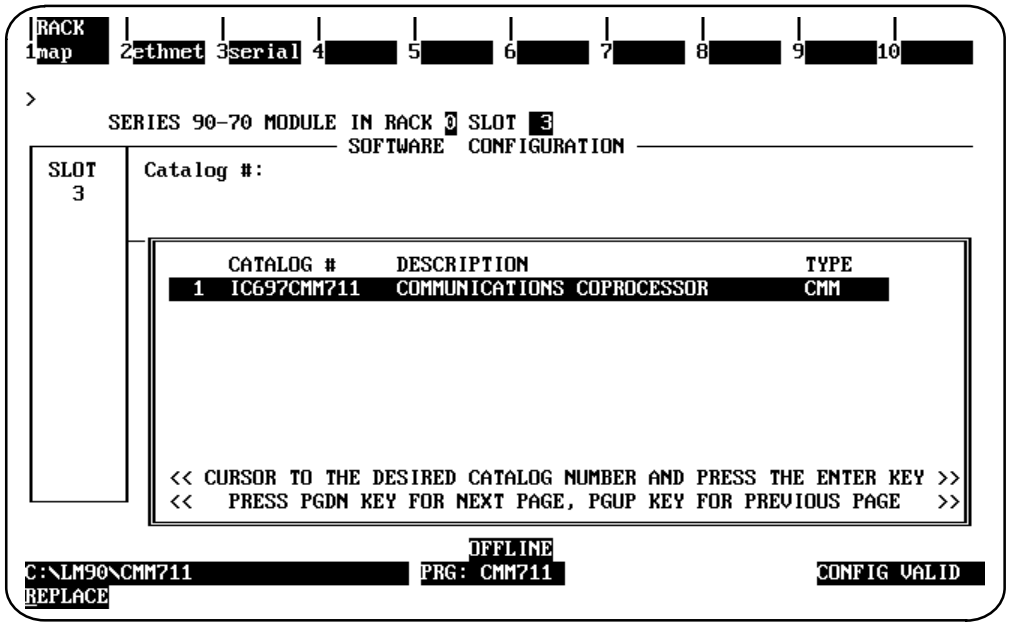
- Pour sélectionner l'emplacement à afficher ou à configurer, utilisez les touches curseur Gauche/Droite.
- Pour afficher un autre rack, utilisez les touches Page Suivante/Précédente ou les touches Haut/Bas.
- Pour ajouter un module, appuyez sur la touche de fonction qui lui correspond.
- Pour changer de rack dans l'API Série 90–70, appuyez sur F5 (rcksel).
- Pour afficher la configuration courante d'un emplacement, appuyez sur F10 (zoom).
- Pour quitter le menu de configuration, appuyez sur Escape.

2.3.1.5. Adjonction d'un module CMM

1. Une fois dans l'écran de configuration du rack d'E/S, sélectionnez le rack et l'emplacement dans lesquels vous voulez installer le module CMM à l'aide des touches curseur.
2. Appuyez sur F6 (comm) pour accéder à l'écran de configuration des modules de communication, illustré ci–après :



3. Appuyez sur F3 (série) pour afficher l'écran qui suit. Dans cet écran, la référence, le nom et le type du module sélectionné s'affichent dans une fenêtre qui se superpose à l'écran précédent, comme illustré ci–dessous :



4. Appuyez sur Enter pour afficher l'écran de configuration du module. Pour plus d'informations, voir le paragraphe Configuration du module ci–après.

2.3.2. Configuration du module (configuration logicielle)

Les paramètres de communication du module CMM peuvent être programmés à l'aide du logiciel de configuration Logicmaster 90. C'est ce que l'on appelle la Configuration logicielle Logicmaster 90.

2.3.2.1. Accès à l'écran de configuration logicielle

1. Pour les modules qui n'ont pas encore été configurés, procédez comme indiqué dans la procédure précédente (Configuration du rack d'E/S).
2. Pour les modules qui sont déjà configurés, positionnez le curseur sur le module désiré et appuyez sur la touche F10 (zoom) dans l'écran de configuration du rack d'E/S.

Si c'est la première fois que vous configurez ce module, l'écran de configuration logicielle CCM SEUL s'affiche. Si le module a déjà été configuré, l'écran de configuration logicielle correspondant au mode précédemment sélectionné s'affiche. Les différents modes de configuration disponibles sont décrits ci–dessous :

2.3.2.2. Modes de configuration du module CMM

Le mode de configuration du module CMM permet de déterminer le protocole de communication qui sera utilisé sur chaque port série du module. Le module CMM peut être configuré de 9 façons différentes :

- CCM SEUL –Protocole CCM sur les deux ports.
- CCM/RTU – Protocole CCM sur port 1, protocole RTU sur port 2.
- RTU/CCM – Protocole RTU sur port 1, protocole CCM sur port 2.
- RTU SEUL – Protocole RTU sur les deux ports.
- SNP SEUL – Protocoles SNP et SNP–X sur les deux ports.
- SNP/CCM – Protocoles SNP et SNP–X sur port 1, protocole CCM sur port 2.
- CCM/SNP – Protocole CCM sur port 1, protocoles SNP et SNP–X sur port 2.
- SNP/RTU – Protocoles SNP et SNP–X sur port 1, protocole RTU sur port 2.
- RTU/SNP – Protocole RTU sur port 1, protocoles SNP et SNP–X sur port 2.

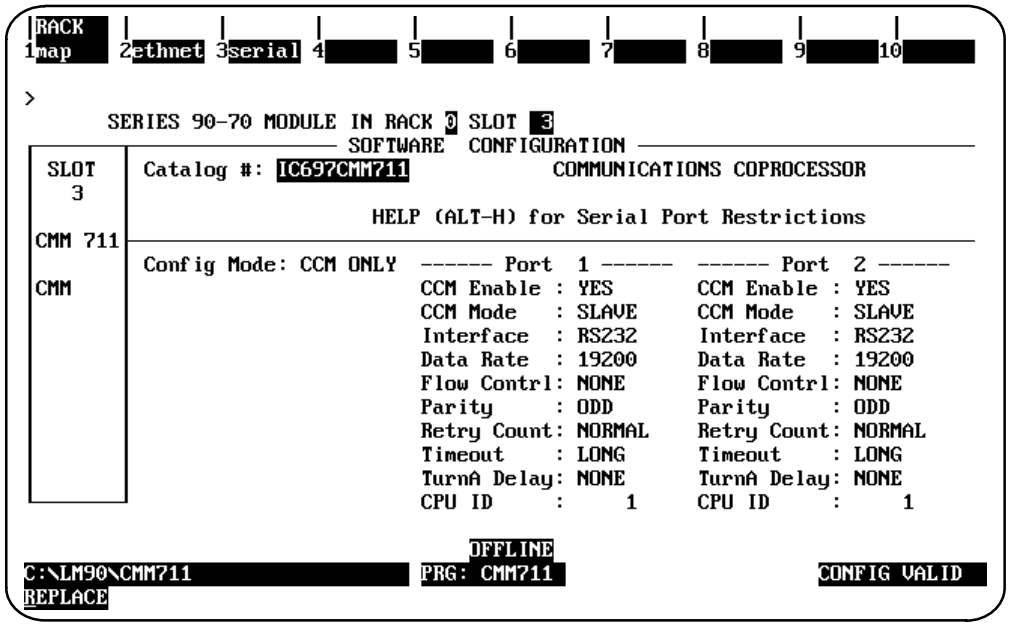
Un écran de configuration logicielle différent s'affiche pour chaque mode de configuration.

2.3.2.3. Modification du mode de configuration

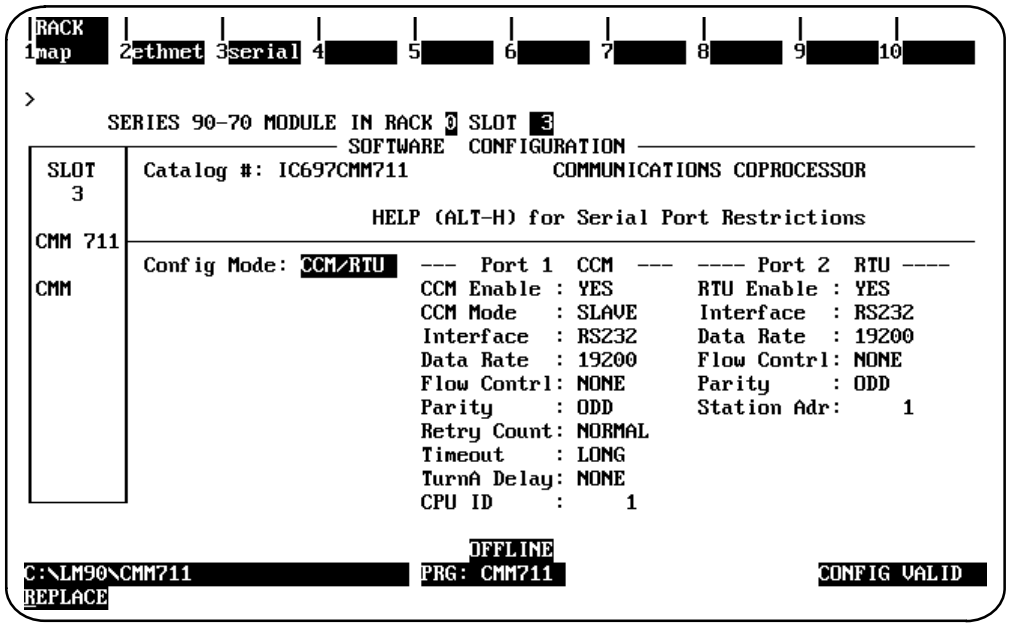
1. A partir de n'importe quel écran de configuration logicielle, positionnez le curseur sur le champ Mode Config.
2. Appuyez sur la touche Tab jusqu'à ce que le mode désiré s'affiche.
3. Appuyez sur la touche Enter pour sélectionner le mode. Les paramètres de communication correspondant au mode sélectionné s'affichent sur l'écran.

Les Figures ci–après montrent les écrans de configuration logicielle des neuf modes de configuration du module CMM :

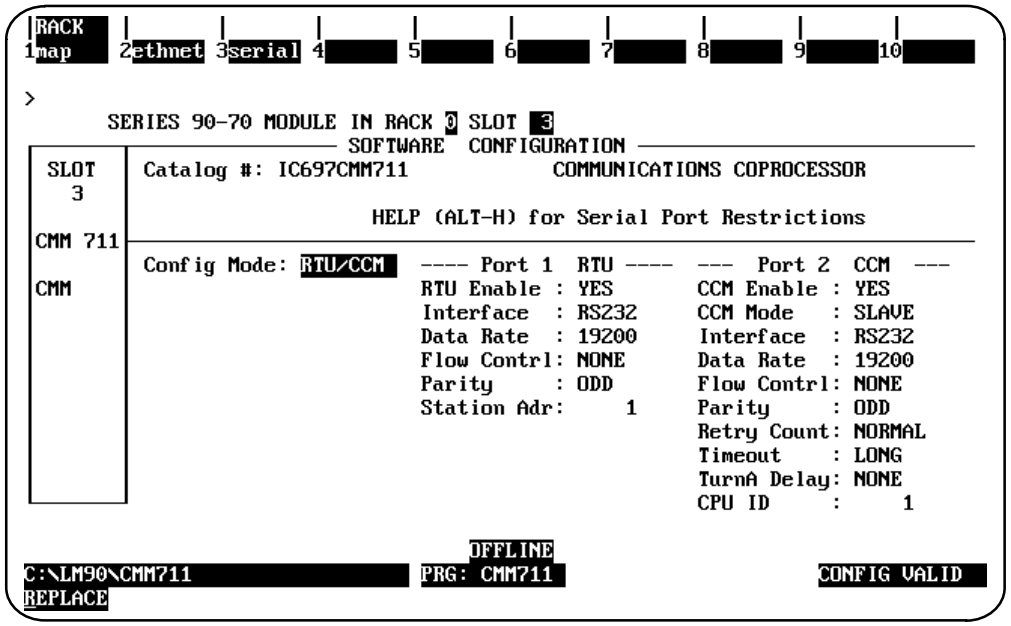
2.3.2.4. Mode CCM SEUL



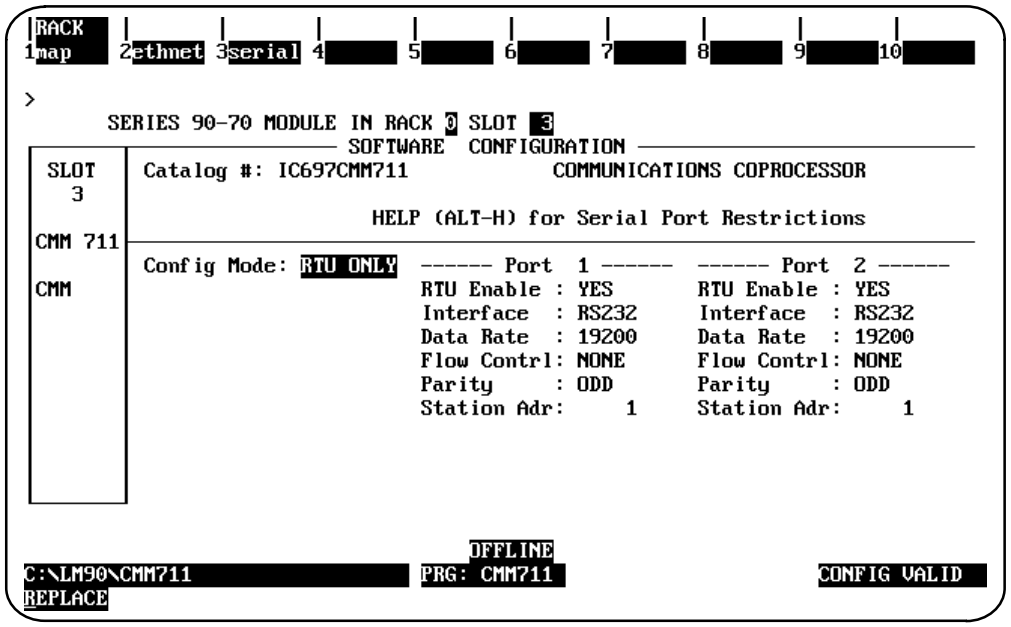
2.3.2.5. Mode CCM/RTU



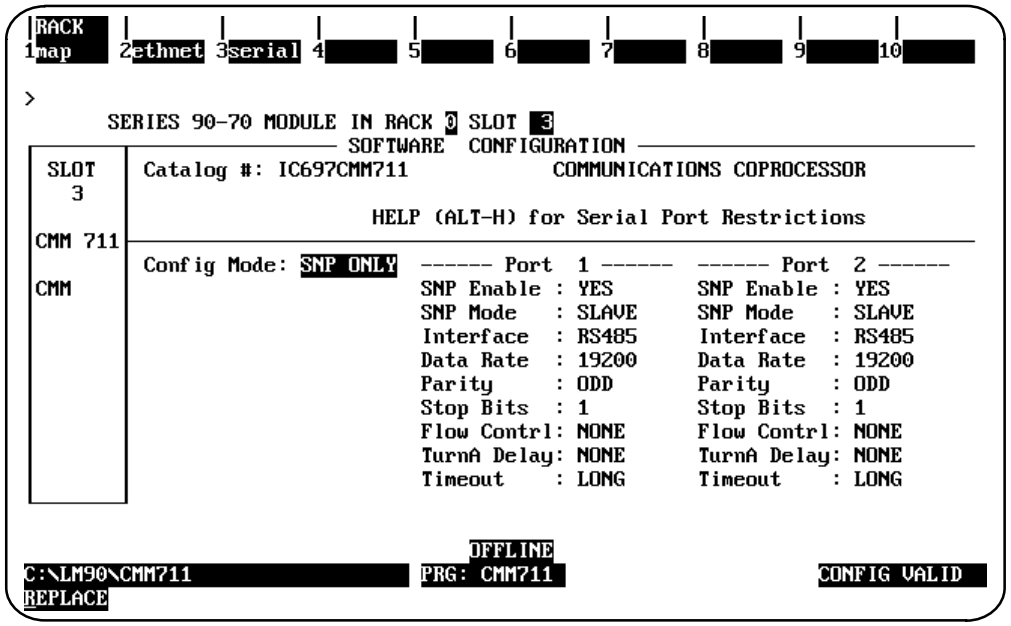
2.3.2.6. Mode RTU/CCM



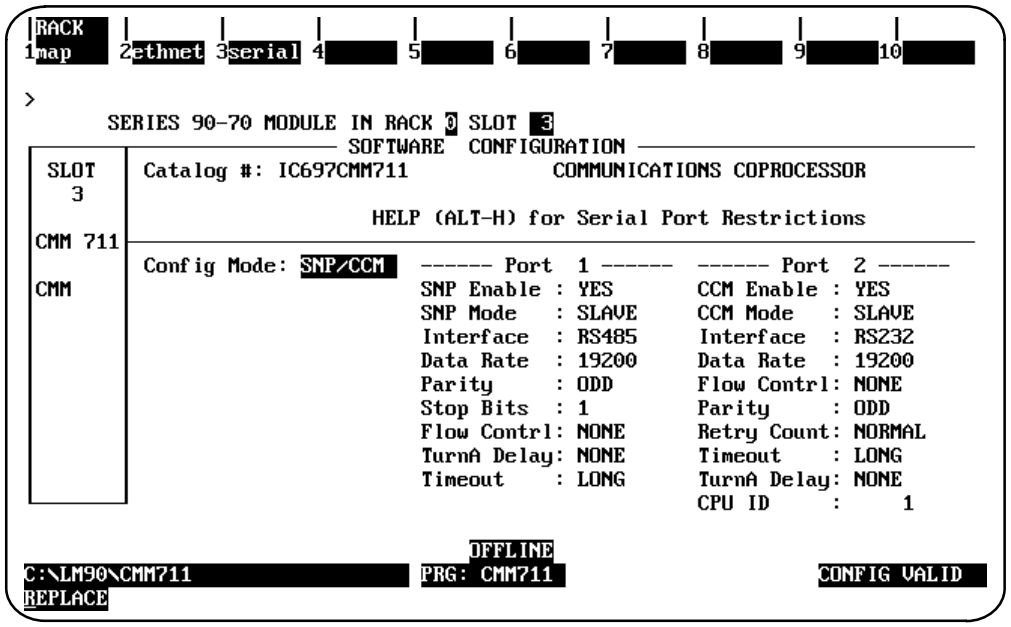
2.3.2.7. Mode RTU SEUL



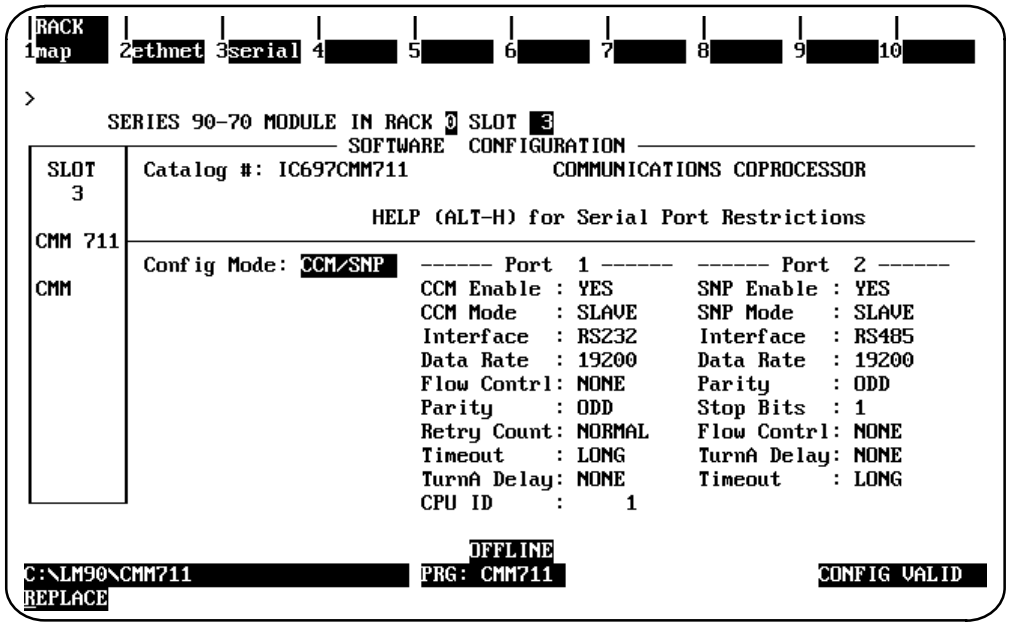
2.3.2.8. Mode SNP SEUL



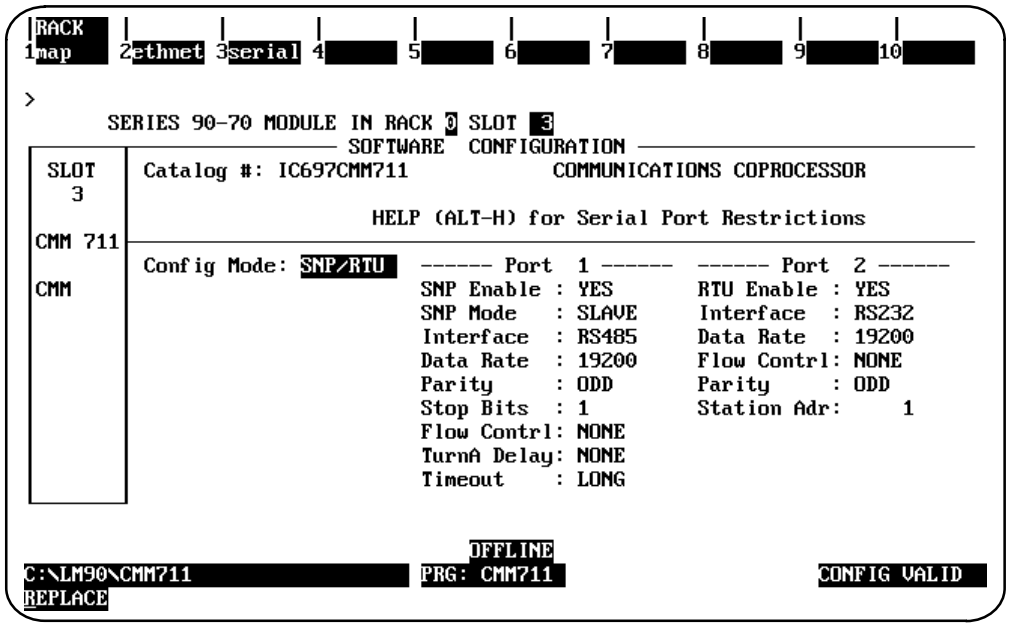
2.3.2.9. Mode SNP/CCM



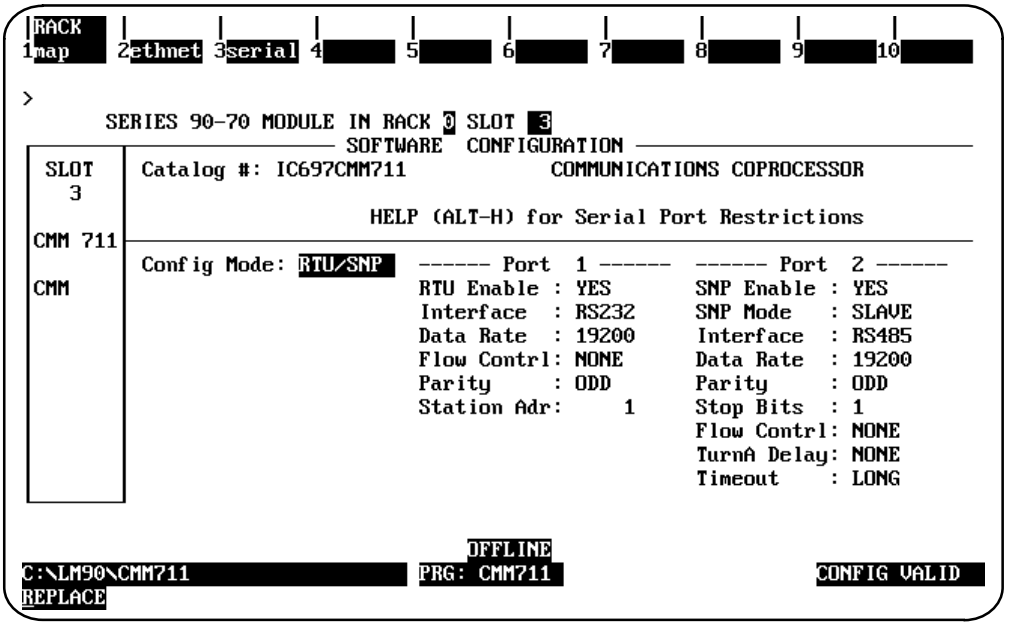
2.3.2.10. Mode CCM/SNP



2.3.2.11. Mode SNP/RTU



2.3.2.12. Mode RTU/SNP



2.3.2.13. Sélection des paramètres de communication

1. A l'aide des touches curseur, positionnez le curseur sur le champ désiré. Les paramètres de communication sont décrits dans le paragraphe qui suit.
2. Sélectionnez la valeur désirée à l'aide de la touche Tab.

2.3.2.14. Enregistrement de la configuration

1. Une fois tous les paramètres de communication sélectionnés, quittez l'écran de configuration logicielle en appuyant sur la touche Escape. L'écran de configuration du rack d'E/S est rafraîchi. Si vous avez correctement configuré le module CMM, celui–ci apparaîtra sur l'écran, dans l'emplacement que vous avez sélectionné.
2. Pour enregistrer les paramètres de communication dans le fichier de configuration, appuyez de nouveau sur la touche Escape.

2.3.2.15. Mémorisation de la configuration dans l'API

Avant de pouvoir travailler en mode connecté avec l'API, vous devez mémoriser la configuration dans l'API à l'aide de la fonction Sauvegarder la configuration du logiciel de configuration Logicmaster 90.

Le message "CONFIG VALIDE" s'affiche dans l'angle inférieur droit de chaque écran, une fois que la configuration a été validée. Si le message indique "CONFIG INVALIDE", le fichier ***n'est certainement pas*** mémorisé dans l'API. Le message CONFIG INVALIDE peut s'afficher dans les cas suivants :

- Un emplacement du rack, avant la carte qui génère des interruptions, a été laissé vacant.
- Les adresses de référence des entrées (%I et %AI) se chevauchent.
- Les modules émetteur/récepteur sur bus BRM/BTM (Bus Transmitter/Receiver Modules) n'ont pas été installés dans l'API Série 90–70.

2.3.2.16. Description des paramètres de communication

Les tableaux ci–dessous décrivent les paramètres de communication disponibles pour chaque protocole série d'un module CMM :

Tableau 3-1. Paramètres de communication CCM

Champ	Choix	Défaut
CCM Validé	OUI, NON	OUI
Mode CCM	ESCLAVE, MAITRE, PAIRE	ESCLAVE
Interface[1]	RS232, RS485	RS232
Vitesse	300, 600, 1200, 2400, 4800, 9600, 19200	19200
Contrl Flux	AUCUN, MATERIEL	AUCUN
Parité	IMPA, SANS	IMPA
Compt Essai	NORMAL, COURT	NORMAL
Temps Limit	LONG, MOYEN, COURT, SANS	LONG
Délai TurnA	SANS, 10 ms, 100 ms, 500 ms	SANS
ID CPU	1–254	1

Tableau 3-2. Paramètres de communication RTU

Champ	Choix	Défaut
RTU Validé	OUI, NON	OUI
Interface[1]	RS232, RS485	RS232
Vitesse	300, 600, 1200, 2400, 4800, 9600, 19200	19200
Contrl Flux	AUCUN, MATERIEL	AUCUN
Parité	IMPA, PAIR, SANS	IMPA
Adr Station	1 – 247	1

Tableau 3-3. Paramètres de communication SNP

Champ	Choix	Défaut
SNP Validé	OUI, NON	OUI
Mode SNP	ESCLAVE, MAITRE	ESCLAVE
Interface[1]	RS485, RS232	RS485
Vitesse	300, 600, 1200, 2400, 4800, 9600, 19200	19200
Contrl Flux	SANS, MATERIEL	SANS
Parité	IMPA, PAIR,SANS	IMPA
Bits Stop	1, 2	1
Temps Limit	LONG, MOYEN, COURT, SANS	LONG
Délai TurnA	SANS, 10 ms, 100 ms, 500 ms	SANS

[1] Pour le module CMM311, le port 1 ne peut être que RS–232.

2.3.2.17. Remarques sur les paramètres de communication

Interface. (CCM, RTU et SNP) Le champ “Interface” spécifie le type d’interface électrique utilisé sur ce port série. Vous avez le choix entre RS–485 et RS–232. Toutefois, pour le module CMM311, le port 1 n’accepte que l’interface RS–232.

Contrôle de flux. (CCM, RTU et SNP) Le champ ”Contrl Flux” spécifie la méthode de contrôle de flux qui sera utilisée sur ce port série. Vous avez le choix entre AUCUN (ou SANS selon le cas) et MATERIEL. Les chronogrammes ci–dessous illustrent l’état des lignes de contrôle de flux pour Contrl Flux = AUCUN (ou SANS) et MATERIEL, respectivement.

AUCUN (ou SANS) utilise les signaux Emission de données (TD) et Réception de données (RD) uniquement. Le signal Demande pour émettre (RTS) est toutefois utilisé comme signal de programmation d’un modem. Le signal RTS excité pendant le laps de temps spécifié pour le délai de retournement du modem (Délai TurnA) et pendant la transmission des caractères ; il est désactivé immédiatement après.

Figure 3-9. Contrl Flux = AUCUN (ou SANS), DELAI TurnA=0

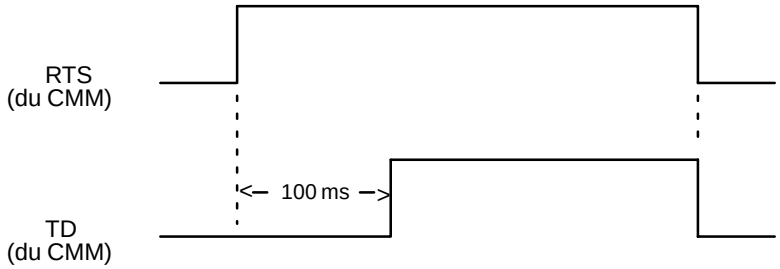


Figure 3-10. Contrl Flux = AUCUN (ou SANS), DELAI TurnA=100 ms

MATERIEL utilise les signaux Emission de données (TD), Réception de données (RD), Demande pour émettre (RTS), Prêt à émettre (CTS), Détecteur du signal de ligne sur la voie des données (DCD), et Equipement terminal de données prêt (DTR). Les signaux sont utilisés comme spécifié dans les normes électriques RS–232 et RS–422/RS–485 ; ces signaux sont décrits ci–après :

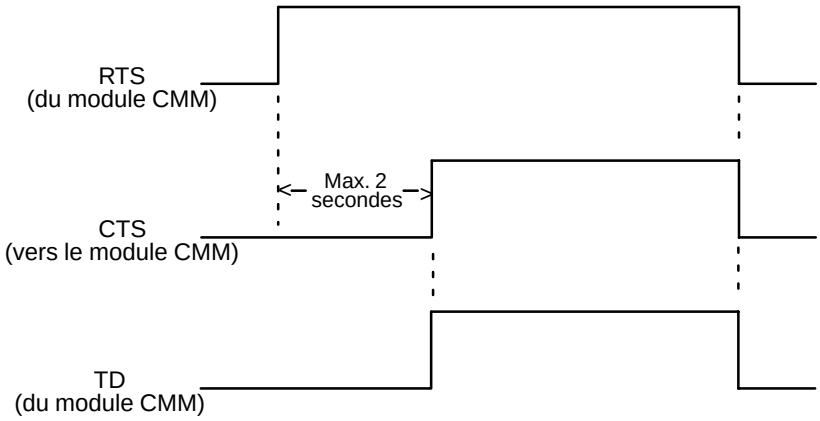


Figure 3-11. Contrl Flux = MATERIEL

Demande pour émettre (RTS) et ***Prêt pour émettre (CTS).*** Ces signaux servent à contrôler le transfert des données sur l'équipement déporté. Le signal RTS est excité au début de chaque transmission par le module CMM. Les caractères ne sont toutefois pas transmis tant que le signal Prêt à émettre (CTS) n'est pas renvoyé. Une fois les caractères transmis, le signal RTS est immédiatement supprimé. Comme dans le cas où il n'y a AUCUN flux de contrôle, tout délai de retournement du modem dont la valeur est différente de zéro n'est PAS utilisé pour commander le signal RTS. Il n'est utilisé que pour régler les temporisateurs appropriés du protocole pour tout retard dans la réception du signal CTS, une fois que le signal RTS a été excité.

Détecteur du signal de ligne sur la voie des données (DCD) et ***Equipement terminal de données prêt (DTR).*** Ces signaux sont utilisés pour contrôler la réception des données depuis l'équipement déporté. Le signal DCD, reçu de l'équipement déporté, constitue essentiellement une demande auprès du module CMM pour qu'il se prépare à recevoir des données. En retour, le module CMM excite le signal DTR dès qu'il est prêt à recevoir des données. Dans le module CMM, le signal DTR est toujours excité ; il n'est jamais désactivé.

Temps Limite. (protocole CCM) Le champ "Temps Limit" spécifie la durée des temporisations utilisées par le protocole CCM sur le port destinataire. Vous avez le choix entre LONG, MOYEN, COURT, ou SANS.

Temps Limite. (SNP) Le champ "Temps Limit" spécifie un jeu de valeurs pour chaque temporisateur SNP. Vous avez le choix entre LONG, MOYEN, COURT, ou SANS. Les choix SNP disponibles et les valeurs auxquelles ils correspondent sont spécifiés dans le tableau qui suit. Les temporisateurs figurant dans la seconde moitié du tableau sont dérivés des précédentes valeurs de temporisation via l'équation spécifiée.

Délai de retournement du modem. (CCM et SNP seulement) Le champ "Délai TurnA" spécifie le laps de temps requis par les modems pour retourner la liaison. Si AUCUN contrôle de flux n'a été sélectionné, le délai de retournement du modem spécifie également le laps de temps pendant lequel le signal Demande pour émettre (RTS) est excité avant la transmission des caractères. Vous avez le choix entre SANS, 10 ms, 100 ms, et 500 ms.

Remarque

Les temporisations utilisées par le logiciel Logicmaster 90 et les CPU Série 90–70 et 90–30 sont également indiquées, à titre de comparaison.

Tableau 3-4. Temporisations SNP

Temporisateur SNP	Sans	Court	Moyen	Long	LM90	90–30	90–70
T1	5 ms	5 ms	5 ms	5 ms	10 ms	5 ms	5 ms
T2	2 s	500 ms	1 s	2 s	3 s	1 s	500 ms
T3’	Désactivé	5 s	10 s	10 s	10 s	10 s	10 s
T4	50 ms	50 ms	50 ms	50 ms	100 ms	N/A	N/A
T5	Désactivé	2 s	5 s	Désactivé	10 s	10 s	10 s
Délai de transmission[1]	0 s	0 s	0 s	0 s	0 s	0 s	0 s
Temps Limit COMREQ[2]	30 s	30 s	30 s	30 s	N/A	N/A	N/A
T3	T3’ – temps de 80 caractères						
T3’’	T3 + 65 secondes						
T5’	T5 + temps de 80 caractères						
T5’’	T5 + Délai de transmission + deux fois le temps de transmission du message qui suit.						

N/A : Non Applicable

Voir Chapitre 7, Définition des protocoles, Temporisateurs SNP, pour une description complète des temporisateurs SNP.

1 Le ”Délai de transmission” spécifie le temps requis pour le transfert des données de l’équipement initiateur à l’équipement répondeur. Ce délai est également appelé ”temps par fil”. Il peut éventuellement être défini à une valeur différente de zéro lorsque les liaisons satellites sont utilisées pour transférer les données par exemple.

2 Le ”Temps Limit COMREQ” spécifie le temps maximal pendant lequel une instruction COMREQ reste valide. Si la commande COMREQ ne peut aboutir avant le temps spécifié, le traitement de COMREQ est interrompu ; un code d’erreur est retourné dans le mot d’état COMREQ pour indiquer que le temps limite de COMREQ a expiré.

2.4. CONFIGURATION DU MODULE CMM311 AVEC LA MINICONSOLE DE PROGRAMMATION HHP

Les paramètres des modules coprocesseur de communication peuvent être modifiés à l'aide de la miniconsole de programmation HHP, pour les CPU version 3 ou ultérieure et les modules CMM version 3.01 ou ultérieure.

2.4.1. Gel de la configuration

Le traitement d'une modification dans la configuration du module CMM prend une quinzaine de secondes. Le temps de traitement de plusieurs modifications simultanées de paramètres est identique à celui d'une seule modification de paramètre. Comme la modification de plusieurs paramètres à la fois est une opération relativement fréquente, les modifications de paramètres individuels sont mémorisées par le module mais ne sont pas traitées et ne prennent pas effet tant que vous ne le demandez pas expressément.

Lorsque vous modifiez un paramètre du module CMM, un astérisque (*) s'affiche devant le nom du module, sur la première ligne de l'écran de la HHP. Il indique que la précédente configuration du module a été "gelée", et que le module n'a pas encore pris en compte la ou les modifications que vous venez de faire. Vous pouvez continuer à modifier des paramètres ; ceux–ci seront mémorisés par le module. Toutefois, si une coupure secteur se produit pendant que la configuration d'un module est gelée, toutes les modifications que vous venez d'effectuer seront perdues.

Lorsque la configuration d'un module est gelée de cette manière, vous pouvez indiquer au système que vous avez fini de modifier les paramètres en appuyant sur les touches WRITE et ENT. Toutes les modifications apportées seront alors traitées simultanément par le module CMM et l'astérisque disparaîtra de l'écran, pour vous signaler que les nouvelles valeurs sont désormais utilisées par le module CMM et ont été enregistrées dans la mémoire non–volatile de l'API.

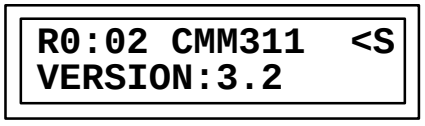
Si entre temps vous renoncez aux modifications que vous venez de faire, vous pouvez les annuler en appuyant sur les touches CLR et ENT. Dans ce cas, tous les paramètres reprendront les valeurs qu'ils avaient avant que la configuration ne soit gelée.

Si vous essayez de sortir de l'emplacement courant (en appuyant sur la touche ↑, ↓, ou #) pendant que la configuration du module est gelée, un message s'affichera pour vous demander si vous désirez utiliser la nouvelle combinaison de valeurs, annuler les nouvelles valeurs et revenir à l'ancienne configuration ou continuer les modifications. Si vous essayez de changer le mode de la console HHP ou de passer en mode RUN, le message d'erreur "FROZEN" (configuration gelée) s'affichera. *Une fois que vous avez modifié des paramètres qui ne sont pas encore pris en compte par le module, vous ne pouvez pas changer d'emplacement tant que les modifications ne sont pas ou enregistrées ou annulées.*

2.4.2. Exemple de modification de la configuration d'un module CMM

Dans cet exemple, nous supposerons qu'un module CMM (IC693CMM311) occupe l'emplacement 2 du rack de la CPU et que l'API a été mis sous tension en appuyant sur les touches CLR et M/T (c'est–à–dire que la mémoire de l'API a été effacée). Dans cet exemple, nous voulons faire passer le mode de CCM ONLY (mode par défaut) à SNP ONLY et programmer la vitesse des deux ports sur 9600.

Affichage initial :



Pour afficher le paramètre Mode :

Appuyez sur la touche

```
R0:02 CMM311  <S
MODE:CCM ONLY
```

Pour afficher successivement tous les modes possibles,

Appuyez sur la touche

```
R0:02 CMM311  <S
MODE:RTU ONLY
```

Chaque fois que vous appuyez sur la touche –/+, un autre mode s'affiche. Lorsque le mode désiré apparaît (affichage clignotant),

Appuyez sur la touche

```
R0:02*CMM311  <S
MODE:SNP ONLY
```

L'astérisque à gauche de CMM indique que la configuration du module est maintenant gelée. C'est–à–dire que le nouveau mode SNP ONLY est mémorisé et affiché, mais que le module fonctionne toujours en mode CCM ONLY (ancienne valeur). Si le module était redémarré à ce stade, le paramètre Mode garderait son ancienne valeur de CCM ONLY.

Si vous essayez de changer le mode de la console HHP ou de passer en mode RUN alors que la configuration du module est gelée, le message d'erreur FROZEN s'affichera. Par exemple :

Appuyez sur la touche

```
R0:02 FROZEN  <S
MODE:SNP ONLY
```

Pour réafficher le nom du module, appuyez sur n'importe quelle touche, par exemple :

Appuyez sur la touche

```
R0:02*CMM311  <S
MODE:SNP ONLY
```

Si à ce stade, vous essayez d'afficher la configuration d'un module installé dans un autre emplacement, un premier message s'affiche pour vous demander si vous souhaitez enregistrer la modification. Par exemple :

Appuyez sur la touche

```
SAVE CHANGES? <S
<ENT>=Y <CLR>=N
```

Comme vous n'avez pas encore modifié la vitesse des ports (comme stipulé dans l'exemple), vous ne voulez pas enregistrer tout de suite vos modifications.

Appuyez sur la touche


```
DISCARD CHGS? <S
<ENT>=Y <CLR>=N
```

Si vous annuliez les modifications à ce stade (comme le suggère le message), vous perdriez la modification de mode déjà effectuée. C'est–à–dire que la configuration reviendrait à CCM ONLY, mode du module avant que la configuration ne soit gelée. Comme vous avez d'autres paramètres à modifier :

Appuyez sur la touche


```
R0:02*CMM311  <S
MODE:SNP ONLY
```

L'astérisque est toujours là pour vous signaler que la configuration du module est toujours gelée et que les modifications effectuées jusque–là ne sont pas encore utilisées par le module. Pour afficher le paramètre Vitesse du port 1,

Appuyez quatre fois sur la touche

```
R0:02*CMM311  <S
DATA RT 1:19200
```

Vous remarquerez que l'astérisque figure toujours à gauche du nom du module. Il indique que la configuration du module est toujours gelée. Vous pouvez continuer à modifier des paramètres, bien qu'aucun d'entre eux ne sera utilisé par le module tant que vous ne les aurez pas enregistrés, comme expliqué ci–dessous. Pour faire passer la vitesse du port 1 à 9600 :

Appuyez successivement sur les touches


```
R0:02 CMM311  <S
DATA RT 1:9600
```

Pour afficher le paramètre Vitesse du port 2 :

Appuyez neuf fois sur la touche

```
R0:02*CMM311  <S
DATA RT 2:19200
```

Pour faire passer la vitesse du port 2 à 9600 :

Appuyez successivement sur les touches


```
R0:02 CMM301  <S
DATA RT 2:9600
```

Pour enregistrer les modifications effectuées :

Appuyez sur la touche


```
SAVE CHANGES? <S
<ENT>=Y <CLR>=N
```

Si à ce stade, vous appuyez sur la touche CLR, l'opération d'enregistrement sera annulée. Puisque vous voulez au contraire que les modifications soient enregistrées,

Appuyez sur la touche


```
PROCESSING     <S
CHANGES
```

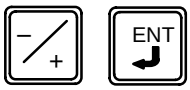
Le mot PROCESSING continuera à clignoter tant que le module n'aura pas fini de traiter les nouvelles valeurs. Ensuite, la console HHP réaffichera le dernier paramètre modifié :

```
R0:02 CMM311   <S
DATA RT 2:9600
```

Vous remarquerez que l'astérisque qui figurait à gauche du nom du module (CMM311) a disparu ; cela signifie que la configuration n'est plus gelée et que le module utilise maintenant les nouvelles valeurs.

Pour reprendre notre exemple, supposons maintenant que vous ayez commencé à modifier les paramètres et que vous constatiez que vous vous êtes trompé. Les modifications effectuées jusque–là (c'est–à–dire depuis que la configuration est gelée), peuvent être annulées de façon à revalider l'ancienne configuration.

Pour le vérifier, faites passer le paramètre Vitesse du port 2 à 4800 :

Appuyez successivement sur les touches 


```
R0:02*CMM311   <S
DATA RT 2:4800
```

Vous remarquerez que la configuration est gelée et que le module CMM311 opère toujours à la vitesse de 9600 (vitesse précédemment programmée).

Pour annuler les modifications,

Appuyez sur la touche


```
DISCARD CHGS? <S
<ENT>=Y <CLR>=N
```

Si à ce stade, vous appuyez de nouveau sur CLR, l'opération d'enregistrement sera annulée.

Appuyez sur la touche


```
R0:02 CMM311   <S
DATA RT 2:9600
```

La configuration du module n'est plus gelée. Les paramètres ont repris les valeurs qu'ils avaient avant que vous ne fassiez passer la vitesse du port 2 à 4800.

Chapitre 4 *Etablissement de la communication – Instruction COMREQ*

Ce chapitre explique comment établir les communications dans les programmes à diagramme en échelle créés à l'aide du logiciel Logicmaster. Il traite des sujets suivants :

1. Demande de communication : Décrit la structure et le principe de fonctionnement de la demande de communication.
2. Instruction COMREQ : Décrit le format et les paramètres de l'instruction COMREQ.
3. Bloc de commande de COMREQ : Décrit le format et le contenu du bloc de commande de COMREQ.

Remarque

Ce chapitre ne concerne que les protocoles CCM, SNP, et SNP–X qui peuvent établir des communications. Le protocole RTU tel qu'il est mis en oeuvre dans le module CMM n'opère qu'en mode esclave et ne peut par conséquent établir de communication.

1. DEMANDE DE COMMUNICATION

L'expression "Demande de communication" sert à décrire tous les éléments utilisateur requis pour établir des communications via le module CMM. La demande de communication utilise les paramètres de *l'instruction COMREQ* et du *bloc de commande* qui lui est associé pour définir les caractéristiques de la demande. Le *mot d'état* enregistre la progression et les résultats de chaque demande.

Ce paragraphe décrit la structure et le principe de fonctionnement de la demande de communication.

1.1. STRUCTURE DE LA DEMANDE DE COMMUNICATION

Au niveau utilisateur, la demande de communication se décompose en trois éléments fondamentaux :

- L'instruction COMREQ
- Le bloc de commande
- Le mot d'état

Le schéma ci–dessous illustre les relations entre ces trois éléments fondamentaux :

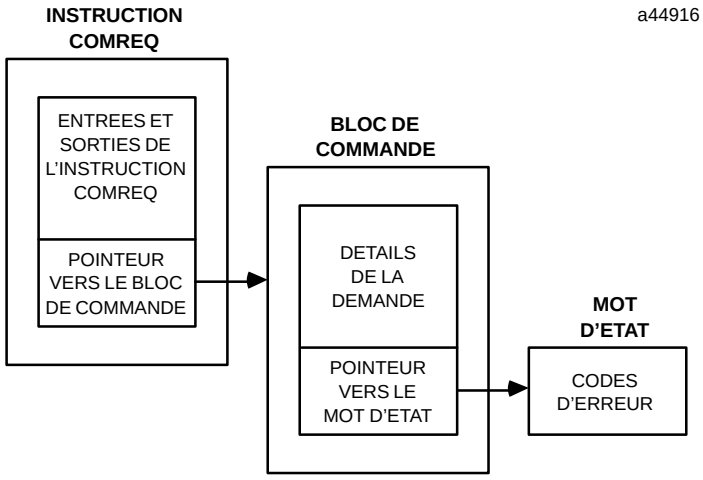


Figure 4-1. Structure de l'instruction COMREQ

L'instruction COMREQ : L'instruction COMREQ est la principale structure utilisée pour spécifier les détails d'une demande de communication à savoir, le rack et l'emplacement du module CMM associé à la demande ainsi que le port sur lequel s'exécutera la demande. De plus, l'instruction COMREQ comporte une entrée qui pointe sur une autre adresse en mémoire : celle du bloc de commande.

Le bloc de commande : Le bloc de commande est une structure à part qui renferme des détails supplémentaires sur la demande de communication : paramètres de synchronisation, pointeur sur le mot d'état et bloc de données. Le bloc de données décrit le sens du transfert de données ainsi que l'adresse et le type de données transférées.

Le mot d'état : Le mot d'état enregistre les résultats de la demande de communication. Pour plus d'informations sur le mot d'état CCM, voir Chapitre 5, Service CCM. Pour plus d'informations sur le mot d'état SNP, voir Chapitre 6, Service SNP.

1.2. PRINCIPE DE FONCTIONNEMENT DE LA DEMANDE DE COMMUNICATION

Le schéma et les paragraphes suivants décrivent ce qui se passe lorsqu'une demande de communication est générée :

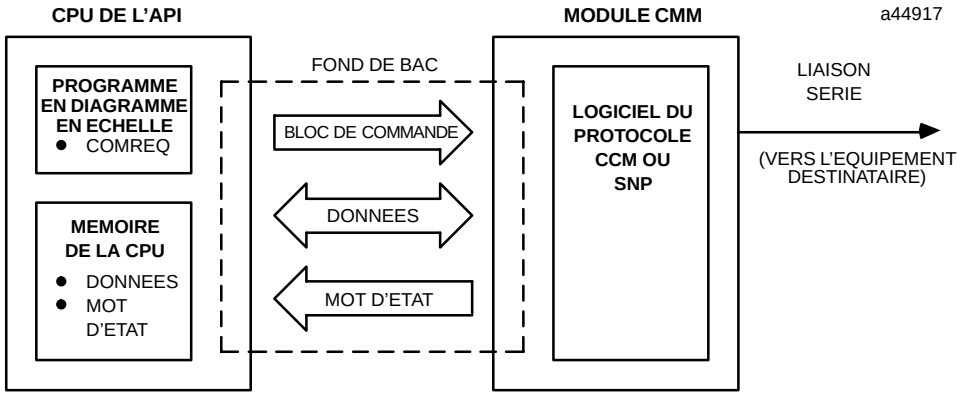


Figure 4-2. Principe de fonctionnement de la demande de communication

1. Une demande de communication est générée chaque fois qu'une instruction COMREQ est activée. A ce stade, les détails de la demande de communication sont transmis de la CPU de l'API au module CMM via un bloc de commande. Le bloc de commande transite par le fond de bac si le module CMM est installé dans un rack local et par les modules d'extension de bus BEM (Bus Expansion Modules pour Série 90–70) et le fond de bac s'il est installé dans un rack d'extension.

2. Certaines demandes sont dites "locales". Elles s'échangent entre la CPU de l'API et le module CMM et ne sont spécifiées que par l'instruction COMREQ. Toutes les communications transitent par le fond de bac ou les modules d'extension de bus BEM (Bus Expansion Modules pour Série 90–70) et le fond de bac. Aucune donnée n'est transmise par la liaison série.

3. Pour les demandes déportées telles que accès en Lecture ou en Ecriture, le module CMM interprète les informations contenues dans le bloc de commande et transmet la demande par la liaison série à l'équipement destinataire (autre module CMM ou hôte) afin qu'elle soit traitée. Les données sont alors transmises de l'équipement destinataire vers l'équipement source (demandes de Lecture) ou de l'équipement source vers l'équipement destinataire (demandes d'Ecriture).

4. A la fin de chaque demande, l'état de la demande est enregistré dans une adresse mémoire particulière indiquée par le pointeur vers le mot d'état spécifié dans le bloc de commande.

1.3. SYNCHRONISATION DU TRAITEMENT DES DEMANDES DE COMMUNICATION

Si le module CMM reçoit des demandes de communication de la CPU de l'API plus vite qu'il ne peut les traiter, il consignera éventuellement un défaut d'application QUEUE FULL (file d'attente pleine) dans la table des défauts automate :

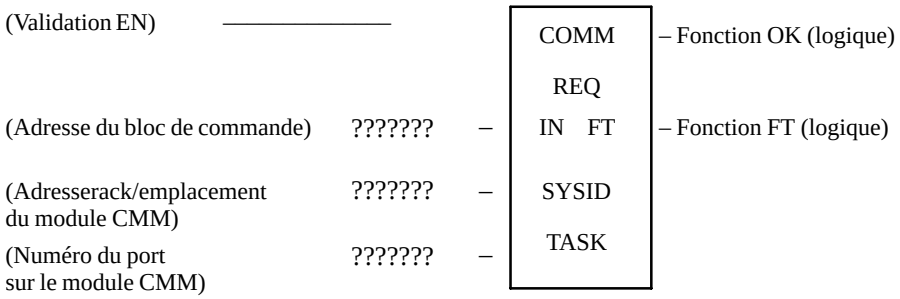
MOD : Other S/W error COMREQ MB FULL START

Ce défaut peut également se produire dans le cas où le module CMM s'arrête de fonctionner.

La première demande de communication transmise à un module CMM après un redémarrage ou un ACFAIL doit être retardée tant que le module CMM n'a pas terminé sa séquence de mise sous tension. La règle générale consiste à attendre 2 secondes après la mise sous tension avant d'essayer de générer une demande de communication.

2. INSTRUCTION COMREQ

La demande de communication démarre lorsque l'instruction COMREQ est activée. L'instruction COMREQ compte quatre entrées et deux sorties :



Chaque entrée et sortie est décrite en détail ci–dessous. Il est important de comprendre que l'entrée Adresse du bloc de commande pointe sur une autre adresse mémoire dans laquelle vous devez entrer des informations complémentaires sur la demande de communication.

Validation EN : Commande logique pour activer l'instruction COMREQ.

IN : Adresse du bloc de commande. Toute adresse valide dans une zone de mémoire orientée Mot (%P, %L, %R, %AI, ou %AQ).

SYSID : Valeur hexadécimale indiquant l'adresse du rack et de l'emplacement du module CMM associé à cette instruction COMREQ.

Si la valeur SYSID programmée correspond à un rack et à un emplacement dans lesquels aucun CMM ou autre module intelligent n'est installé, la sortie Fonction OK ne sera pas activée.

Exemples :

Rack	Emplacement	Valeur Hex. du mot
0	4	0004h
3	4	0304h
2	9	0209h
7	2	0702h

TASK : Port série du module CMM qui sera utilisé par cette instruction COMREQ.

Le tableau ci–dessous indique les numéros de tâches (TASK) valides pour le module CMM :

Numéro de tâche	Description
1	Protocole du Port 1
2	Protocole du Port 2

Si le numéro de tâche programmé pour le module CMM n'est pas valide, un défaut d'application sera consigné dans la table des défauts : COMREQ BAD TASK ID. Cette entrée sera consignée si la tâche spécifiée dans l'instruction COMREQ est mal programmée ou si le module CMM a été configuré sans qu'aucun protocole de communication n'ait été validé sur ce port série.

Sorties OK et FT : La sortie des fonctions OK et FT (défaut) peut fournir un flux validant aux instructions optionnelles, ce qui permet de vérifier que la demande de communication a abouti. Les sorties OK et FT peuvent prendre les états suivants :

Validation (EN)	Erreur ?	Sortie OK	Sortie FT
actif	non	vrai	faux
actif	oui	faux	vrai
non actif	pas d'exécution	faux	faux

Flux validant : L'instruction COMREQ génère toujours un flux validant sur la sortie OK en mode NOWAIT. En mode WAIT, la fonction génère un flux validant sur la sortie OK, sauf si le temps limite est dépassé ou si un temps limite nul (0) a été spécifié. La sortie OK est alors mise à faux et la sortie FT à vrai. Les modes WAIT/NOWAIT sont décrits en détail dans le paragraphe 3 : Bloc de commande de COMREQ, du présent chapitre.

La sortie FT peut également être mise à Vrai et la sortie OK à Faux dans les cas suivants :

- L'adresse destinataire spécifiée n'existe pas.
- La tâche spécifiée (TASK) n'est pas valide pour l'équipement.
- La longueur des données a été configurée à 0.

S'il y a des erreurs dans la partie du bloc de commande utilisée de façon spécifique par le module CMM, ces erreurs apparaîtront dans la valeur retournée au mot d'état, et non pas au niveau de la sortie FT.

3. BLOC DE COMMANDE DE COMREQ

Le bloc de commande renferme les détails d'une demande de communication. L'adresse du bloc de commande est spécifiée dans l'entrée IN de l'instruction COMREQ. Elle correspond généralement à une zone de mémoire orientée Mot (%P, %L, %R, %AI, ou %AQ). La structure du bloc de commande peut être placée dans la zone de mémoire spécifiée à l'aide de l'instruction de programmation appropriée (l'utilisation de l'instruction de déplacement de bloc BLOCK MOVE est recommandée).

La structure du bloc de commande est la suivante :

Longueur du bloc de données	Mot 1
Indicateur WAIT/NOWAIT	Mot 2
Type de mémoire pour mot d'état	Mot 3
Décalage du pointeur d'état	Mot 4
Temps limite d'inactivité	Mot 5
Durée maximale de la communication	Mot 6
Bloc de données	Mots 7 jusqu'à 134

Lorsque vous définissez le bloc de commande, reportez–vous aux définitions ci–dessous :

Longueur du bloc de données (Mot 1) : Longueur de la partie bloc de données du bloc de commande (de 1 à 128 mots). Les données commencent au mot 7 du bloc de commande. La longueur doit être spécifiée à partir du mot 7 et non pas à partir du début du bloc de commande.

Indicateur WAIT/NOWAIT (Mot 2) : Cet indicateur détermine si le programme de l'API Série 90 attendra que le module CMM ait reçu les données annoncées avant de poursuivre. Il peut prendre deux valeurs :

0 = n'attendra pas (NOWAIT)

1 = attendra une réponse (WAIT)

La demande peut consister à envoyer un message et à attendre une réponse, ou à envoyer un message et à poursuivre sans attendre de réponse. S'il est spécifié dans le bloc de commande que le programme ***n'attendra pas la réponse,*** le contenu du bloc de commande sera transmis au module CMM et l'exécution du programme en diagramme en échelle reprendra immédiatement. C'est ce que l'on appelle le mode NOWAIT.

S'il est spécifié dans le bloc de commande que le programme ***attendra la réponse***, le contenu du bloc de commande est transmis au module CMM et la CPU attend une réponse. Le temps limite maximal pendant lequel l'API attendra que l'équipement réponde est spécifié dans le bloc de commande. Si l'équipement ne répond pas dans le temps imparti, l'exécution du programme en diagramme en échelle reprendra. C'est ce que l'on appelle le mode WAIT.

Attention

Avec le protocole CCM, il est recommandé de programmer cet indicateur sur NOWAIT. Sinon, le temps passé par le protocole CCM à attendre risque d'avoir un effet négatif sur le cycle de la CPU.

Avec le protocole SNP, cet indicateur *doit* obligatoirement être programmé sur NOWAIT. Toute instruction SNP COMREQ avec l'indicateur WAIT sera immédiatement retournée avec un code d'erreur.

Type de mémoire pour mot d'état (Mot 3) : Le mot d'état est mémorisé dans la CPU de l'API dans l'adresse spécifiée par le Mot 3 et le Mot 4 du bloc de commande. Le format de cette adresse indique le type de mémoire (Mot 3) et le décalage (Mot 4).

Abréviation	Type de mémoire	Valeur à entrer	
		Décimal	Hexadécimal
%I	Table des entrées logiques	70	46h
%Q	Table des sorties logiques	72	48h
%R	Mémoire des registres	8	08h
%AI	Table des entrées analogiques	10	0Ah
%AQ	Table des sorties analogiques	12	0Ch

Décalage du pointeur d'état (Mot 4) : Ce mot renferme la valeur de décalage dans le type de mémoire sélectionné. ***Le décalage du pointeur vers le mot d'état démarre à zéro.*** Pour indiquer par exemple %R1 comme adresse de mot d'état, vous devez indiquer un décalage nul (0). De la même manière, le décalage correspondant à %R100 serait 99.

Pour plus d'informations sur le contenu du mot d'état CCM, voir Chapitre 5, Service CCM.

Pour plus d'informations sur le contenu du mot d'état SNP, voir Chapitre 6, Service SNP.

Temps limite d'inactivité (Mot 5) : La valeur spécifiée correspond au laps de temps maximal pendant lequel l'API attendra un accusé de réception de la demande de la part du module CMM. En mode NOWAIT, cette valeur n'est pas utilisée. (Toute valeur spécifiée sera ignorée ; elle peut être définie à zéro). En mode WAIT, ce mot spécifie le temps limite d'inactivité en incréments de 100 microsecondes.

Durée maximale de la communication (Mot 6) : Ce mot indique le laps de temps maximal pendant lequel le programme "gardera la fenêtre ouverte" pendant que le module CMM est actif. En mode NOWAIT, cette valeur n'est pas utilisée. (Toute valeur spécifiée sera ignorée ; elle peut être définie à zéro). En mode WAIT, ce mot spécifie la durée maximale de la communication en incréments de 100 microsecondes.

Bloc de données (Mots 7 – 134) : Le bloc de données renferme des informations sur la demande sous un format qui varie suivant le protocole de communication utilisé.

Pour plus d'informations sur le bloc de données sous le protocole CCM, voir Chapitre 5, Service CCM. La longueur du bloc de données CCM est limitée à 6 mots maximum.

Pour plus d'informations sur le bloc de données sous le protocole SNP, voir Chapitre 6, Service SNP. La longueur du bloc de données SNP dépend de la commande SNP utilisée.

Chapitre 5 Service CCM

Ce chapitre décrit le service de communication série CCM. Pour bien comprendre, il est important d'avoir lu le Chapitre 4, Etablissement de la communication – instruction COMREQ.

Ce chapitre traite des sujets suivants :

- 1 : Le bloc de données CCM COMREQ
- 2 : Le mot d'état CCM COMREQ
- 3 : Exemples de programmation CCM COMREQ

Pour les utilisateurs qui connaissent le fonctionnement du protocole CCM sur les automates programmables GE Fanuc, des tableaux récapitulatifs du protocole sur des API Série 90 sont décrits en Annexe C.

1. LE BLOC DE DONNÉES CCM COMREQ

Le bloc de données CCM COMREQ est la partie du bloc de commande COMREQ qui identifie la commande CCM à exécuter et fournit les paramètres de cette commande.

Ce paragraphe traite des sujets suivants :

- Structure du bloc de données CCM
- Résumé du bloc de données pour les commandes CCM
- Types de mémoire CCM
- Adressage de la mémoire CCM et Longueur des données

1.1. STRUCTURE DU BLOC DE DONNÉES CCM

La structure du bloc de données CCM est la suivante :

Description	Mot du bloc de commande
Mot de commande CCM	Mot 7
ID destinataire	Mot 8
Type de mémoire destinataire	Mot 9
Adresse mémoire destinataire	Mot 10
Longueur des données	Mot 11
Adresse mémoire source	Mot 12

Chaque élément du bloc de données CCM est décrit ci–dessous :

Numéro de la commande CCM (Mot 7) : Ce mot identifie la commande CCM qui sera exécutée. Il existe deux catégories de commandes CCM : les commandes locales et les commandes déportées. Les commandes locales sont destinées à la CPU à l'origine de la demande et au module CMM qui lui est associé. Elles ne transitent pas par la liaison série. Les commandes déportées, telles que Accès en Lecture/Ecriture, sont transmises par la liaison série. La liste complète des commandes CCM est fournie plus loin dans ce paragraphe, sous la rubrique Résumé du bloc de données pour commandes CCM.

ID destinataire (Mot 8) : Pour pouvoir transférer des données entre des équipements CCM, l'un des équipements CCM doit demander le transfert et l'autre doit satisfaire la demande. L'équipement qui demande ou initialise le transfert est appelé l'équipement ***source*** ; celui qui le satisfait, mais ne l'initialise pas est appelé l'équipement ***destinataire***. Les données peuvent circuler de l'équipement source à l'équipement destinataire, tout comme de l'équipement destinataire à l'équipement source.

L'ID destinataire est le numéro d'identification de l'équipement destinataire ; pour un CCM Série 90, cela correspond à l'ID CPU. Chaque port CCM peut être configuré sur le même ID CPU ou sur des ID CPU distincts. L'ID CPU est affecté à l'aide du logiciel de configuration Logicmaster 90. Pour plus d'informations à ce sujet, voir le Chapitre 3. L'ID CPU 1 est automatiquement affecté aux deux ports par la configuration par défaut fournie par la CPU modèle 331 d'un API Série 90–30. L'ID CPU 1 est également la valeur de configuration par défaut initialement fournie par le logiciel Logicmaster 90.

L'ID destinataire est un nombre compris entre 1 et 255 en mode Egal à égal, ou entre 1 et 90 en mode maître–esclave. L'ID destinataire 0 est réservé. *En mode Egal à égal, tout équipement CCM répondra à l'ID destinataire 255, quel que soit son ID.*

Type de mémoire destinataire (Mot 9) : Type de mémoire auquel le bloc de données accède dans l'équipement CCM destinataire. Neuf types de mémoire destinataire Série 90 sont accessibles (1, 2, 3, 6, 9, 13, 14, 17, et 18). Les différents types de mémoire sont décrits plus loin dans ce paragraphe, sous la rubrique Types de mémoire CCM.

Adresse mémoire destinataire (Mot 10) : L'adresse mémoire destinataire spécifie l'adresse dans l'équipement CCM destinataire où le transfert de données doit commencer. La gamme d'adresses de chaque type de mémoire Série 90 ainsi que des exemples d'adressage sont fournis plus loin dans ce paragraphe, sous la rubrique Adressage de la mémoire CCM et Longueur des données.

Remarque

Au niveau du type de mémoire et de l'adresse mémoire destinataire, le contrôle d'erreur est effectué par l'équipement récepteur, et non pas par le module CMM qui établit la demande. Certains types et adresses mémoires destinataires qui ne sont pas valides pour la CPU qui établit la demande, peuvent être valides pour l'équipement CCM qui répond.

Longueur des données (Mot 11) : Longueur du transfert de données. Les unités sont déterminées par le type de mémoire de l'équipement source, spécifié par le numéro de la commande. Les unités de longueur et les incréments de chaque type de mémoire sont décrits à l'aide d'exemples plus loin dans ce paragraphe, sous la rubrique Adressage de la mémoire CCM et Longueur des données.

Adresse mémoire source (Mot 12) : L'adresse mémoire source spécifie l'adresse de la CPU Série 90 où le transfert de données doit commencer. La gamme d'adresses de chaque type de mémoire Série 90 ainsi que des exemples d'adressage sont fournis plus loin dans ce paragraphe, sous la rubrique Adressage de la mémoire CCM et Longueur des données.

1.2. RÉSUMÉ DU BLOC DE DONNÉES POUR LES COMMANDES CCM

Le premier mot du bloc de données (Mot 8 du bloc de commande) doit être une commande, dans la gamme 6000 à 6199 (décimal).

La sous–gamme 6000 à 6099 est réservée à des fonctions à caractère général qui ne concernent que le stockage des données locales dans le module CMM (commandes locales) et qui sont généralement utilisées dans tous les modes du CCM (maître, esclave, récepteur et émetteur en mode Egal à égal).

La sous–gamme 6100 à 6199 est utilisée pour les opérations qui établissent une communication via la liaison série (commandes déportées) et son usage est généralement limité aux modes CCM maître et Emetteur–Egal à égal.

Le tableau ci–dessous liste les mots de commande et les paramètres requis par chacun d'eux. Pour plus d'informations et des exemples de chaque commande, voir paragraphe 3 du présent chapitre.

Tableau 5-1. Résumé du bloc de commande COMREQ pour les commandes CCM

Description de la commande	Taille du bloc de données	Registres du bloc de données (X pour "Obligatoire" ; — pour "Non utilisé")					
		Mot de commande Mot 7	**ID destinataire Mot 8**	**Type mémoire destinataire Mot 9**	**Adresse mémoire destinataire Mot 10**	**Longueur des données Mot 11**	**Adresse mémoire source Mot 12**
Programmer Réponse Q[1,2] (mode esclave seul)	3 mots	6001 (1771h)	X[3a]	X[3b]	—	—	—
Effacer mots d'état de diagnostic CCM[1]	1 mot	6002 (1772h)	—	—	—	—	—
Lire mots d'état de diagnostic CCM [1] dans registres source	6 mots (dont 2 inutilisés)	6003 (1773h)	—	—	X	X	X
Configuration logicielle[1]	15 mots	6004 (1774h)	Pour plus de détails, voir paragraphe 3				
Lire Table des registres de Destinataire à Source	6 mots	6101 (17D5h)	X	X	X	X	X
Lire Table des entrées de Destinataire à Source	6 mots	6102 (17D6h)	X	X	X	X	X
Lire Table des sorties de Destinataire à Source	6 mots	6103 (17D7h)	X	X	X	X	X
Lire Réponse Q dans table des registres Source	6 mots (3 utilisés)	6109 (17DDh)	X	—	—	—	X[4]
Ecrire un seul bit	4 mots	6110 (17DEh)	X	X	X	—	—
Ecrire Table des registres Source dans Destinataire	6 mots	6111 (17DFh)	X	X	X	X	X
Ecrire Table des entrées Source dans Destinataire	6 mots	6112 (17E0h)	X	X	X	X	X
Ecrire Table des sorties Source dans Destinataire	6 mots	6113 (17E1h)	X	X	X	X	X

1 Commande interne (pas de communication par le port série).

2 La Réponse Q peut être programmée via la commande 6001, et le CCM esclave répondra à une demande de renseignement Q reçue d'un équipement externe sur le port externe. Pour plus d'informations sur la séquence Q, voir paragraphe 3 du présent chapitre.

3a Octets de données 1 et 2.

3b Octets de données 3 et 4.

4 Adresse dans mémoire des registres (%R) de l'équipement source.

1.3. TYPES DE MÉMOIRE CCM

Les tableaux ci–dessous listent les types de mémoire supportés par le protocole CCM Série 90. Les types de mémoire pour la mise à 1 et à 0 de bits CCM uniques sont des types de mémoire logiques. Ils pointent sur les mêmes tables des entrées et des sorties que les types de mémoire 1 et 2, mais ont des numéros uniques dans la mesure où ils sont utilisés pour des opérations spéciales de mise à 1 et à 0 de bits dans les tables des entrées et des sorties.

Tableau 5-2. Types de mémoire supportés par le protocole CCM Série 90

Type de mémoire CCM	Tables CCM destinataire	Opération
1	Table des registres	Lecture/Ecriture
2	Table des entrées	Lecture/Ecriture
3	Table des sorties	Lecture/Ecriture
6	Zone de travail CCM	Lecture
9	Mots d'état de diagnostic	Lecture
13	Tables des entrées	Mise de bits à 1
14	Table des sorties	Mise de bits à 1
17	Table des entrées	Mise de bits à 0
18	Table des sorties	Mise de bits à 0

1.3.1. Zone de travail CCM (Type de mémoire 6)

La zone de travail est entièrement mise à jour chaque fois qu'une demande READ (Lecture) externe est reçue par le CCM. Toutes les adresses de la zone de travail sont de type *lecture seule*. La zone de travail est un type de mémoire orienté Octet.

Tableau 5-3. Affectation de la mémoire de zone de travail CCM

Adresse SP	Identificateur de champ	Bits							
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00	Etat Run CPU	0	0	0	0	Voir Notes (1)			
01	Etat Commande CPU	Configuration binaire identique à SP(00)							
02 03	Type CPU	Majeur(2a) (en hexadécimal) Mineur(2b) (en hexadécimal)							
04–0B	ID CPU SNP	7 caractères ASCII + caractère de fin = (00h)							
0C 0D	No. Révision Microprogramme CPU	Majeur (en BCD) Mineur (en BCD)							
0E 0F	No. Révision Microprogramme PCM/CMM	Majeur (en BCD) Mineur (en BCD)							
10–11	Réservées	(00h)							
12	ID Type de noeud	(90–70 : 0Ch ; 90–30/35 : 0Dh)							
13–15	Réservées	(00h)							
16	ID CPU CCM	Maître/esclave : 1–90 (Décimal) Egal à égal : 1–254							
17	Réservée	(00h)							
18–33	Tailles des types de mémoire	Voir Note (3)							
18–1B	Mémoire des registres	Taille %R							
1C–1F	Table des entrées analogiques	Taille %AI							
20–23	Table des sorties analogiques	Taille %AQ							
24–27	Table des entrées	Taille %I							
28–2B	Table des sorties	Taille %Q							
2C–2F	Mémoire logique interne	Taille %M							
30–33	Code Programme Utilisateur	Voir Note (4)							
34–FF	Réservées	(00h)							

1.3.1.1. Notes relatives aux affectations de la mémoire de zone de travail

1 0000 = Run_Validé
0001 = Run_Invalidé
0010 = Stoppé
0011 = Stoppé_En_défaut
0100 = Arrrêté
0101 = Suspendu
0110 = Stoppé_ES_Validées

2a Types majeurs des CPU :
12 (0Ch) CPU API Série 90–70
16 (10h) CPU API Série 90–30

2b Types mineurs des CPU Série 90–70 :
31 (1Fh) CPU 731
32 (20h) CPU 732
71 (47h) CPU 771
72 (48h) CPU 772
81 (51h) CPU 781
82 (52h) CPU 782
88 (58h) CPU 788
89 (59h) CPU 789
92 (5ch) CPU 914
94 (5eh) CPU 924

Sous–types Série 90–30 :
35 (23h) CPU 331
36 (24h) CPU 341

3 Octets 18h–33h de la zone de travail :

Octets	Longueur de la mémoire		Taille retournée en
18–1B	%R	Mémoire des registres	Mots
1C–1F	%AI	Table des entrées analogiques	Mots
20–23	%AQ	Table des sorties analogiques	Mots
24–27	%I	Table des entrées	Points (Bits)
28–2B	%Q	Table des sorties	Points (Bits)
2C–2F	%M	Mémoire logique interne	Points (Bits)
30–33		Code Programme Utilisateur	Octets

Remarque

La longueur en hexadécimal de chaque type de mémoire est codée sur quatre octets dont le mot de poids fort est réservé pour un usage ultérieur. Par exemple, la taille par défaut de 1024 mots (0400h) de la mémoire des registres de la CPU 731 serait retournée sous le format suivant :

Mot	Moins significatif		Plus significatif	
Octet SP	18	19	1A	1B
contient	00	04	00	00

4 Espace de mémoire programme occupé par le programme. S'affiche également dans l'écran Mémoire utilisée de l'API Logicmaster 90 dans le champ Prog Utilisateur.

1.3.2. Mots d'état de diagnostic

Outre le mot d'état CCM qui est automatiquement transféré du module CMM à la CPU, il existe 20 mots d'état de diagnostic créés et mis à jour dans le module CMM. Les mots d'état de diagnostic ne sont pas automatiquement transmis à la CPU ; la commande COMREQ 6003 interne (Lire mots d'état de diagnostic dans registres source) sert à transférer ces mots d'état du module CMM à la CPU. Un équipement externe peut accéder à ces mots d'état à l'aide d'une instruction READ et le type de mémoire destinataire 9. Le tableau ci–dessous explique le rôle de chaque mot d'état de diagnostic.

Lorsque deux ports CCM fonctionnent en même temps, chacun dispose de sa propre copie des mots d'état de diagnostic. Aucun des deux ne peut indiquer l'état de l'autre.

Les API Série 90 disposent de deux jeux distincts de mots d'état de diagnostic, comme illustré dans le tableau ci–dessous.

Tableau 5-4. Définition des mots d'état de diagnostic CCM

Mot d'état de diagnostic	Contenu du mot	
	Octet 2	**Octet 1 (BPf)**
1	00h	Code d'erreur du port série(1)
2	Nombre de conversations ayant abouti(2)	
3	Nombre de conversations arrêtées prématurément(2)	
4	Nombre de retransmissions de l'en–tête	
5	Nombre de retransmissions du bloc de données	
6	Nombre de séquences Q ayant abouti	
7	Nombre de collisions d'égal à égal	
8–11	Réservés (00h)	
12	No Version du logiciel PCM/CMM(3)	
13	Mot d'état COMREQ(4)	
14	Réservé (00h)	
15–20	Contenu du bloc de données COMREQ	

1 Voir la liste complète des codes d'erreurs et leur définition dans le paragraphe 2 de ce chapitre.

2 Les commandes internes ne modifieront pas ce comptage. Le terme "conversation" désigne une communication série via le port série.

3 Identique au numéro de révision du microprogramme PCM/CMM dans la zone de travail (0E–0F). Cette valeur restera toujours dans le mot 12 des mots d'état de diagnostic, même lorsque les mots d'état de diagnostic sont effacés à l'aide de la commande interne 6002 ou à la demande d'un équipement externe.

4 Voir la description du mot d'état retourné pour un CCM COMREQ dans le paragraphe 2 du présent chapitre.

1.4. ADRESSAGE DE LA MÉMOIRE CCM ET LONGUEUR DES DONNÉES

Pour pouvoir procéder à un transfert de données, le protocole CCM doit connaître l'adresse à laquelle le transfert commencera et la longueur des données à transférer. L'adresse de départ plus la longueur ne doivent pas excéder la limite de la table concernée. Les méthodes de spécification de l'adresse de départ et de la longueur des données sont décrites dans ce paragraphe.

1.4.1. Adresses mémoire CCM

Les adresses mémoire figurant dans le tableau ci–dessous sont les adresses destinataires lorsque l'équipement *qui répond* est un API Série 90 et les adresses source lorsque l'équipement *qui demande la communication* est un API Série 90.

Tableau 5-5. Adresses mémoire destinataire/source

Type de mémoire		Description	Gamme d'adresses(1)
1	Registre	Spécifie le registre à partir duquel le transfert de données doit commencer.	1–Maximum
2 3	Table des entrées Table des sorties	Spécifie le point d'entrée ou de sortie à partir duquel le transfert de données doit commencer. L'adresse mémoire source doit commencer sur un octet (exemple, 1, 9, 17 ...).(2)	1–Maximum
6	Mémoire de la zone de travail CCM(3)	Spécifie l'octet de la zone de travail à partir duquel le transfert de données doit commencer.	0–255
9	Mots d'état de diagnostic CCM (3)	Spécifie le mot d'état de diagnostic à partir duquel le transfert de données doit commencer.	1–20
13 14	Entrée Bit à 1 Sortie Bit à 1	Spécifie le point d'entrée ou de sortie à mettre à 1.	1–Maximum
17 18	Entrée Bit à 0 Sortie Bit à 0	Spécifie le point d'entrée ou de sortie à mettre à 0.	1–Maximum

1 Les gammes maximales adressables pour chaque type de mémoire dépendent du modèle de la CPU et de la configuration de la mémoire.

2 Pour les références E/S, les mises en oeuvre du protocole CCM Série 90 utilisent un adressage orienté Point. L'adresse de départ est interprétée par l'API Série 90 comme le numéro de bit à partir duquel le transfert des données doit commencer. Les adresses mémoire source Série 90 doivent commencer sur un octet. (Voir les exemples qui suivent).

3 La zone de travail et les mots d'état de diagnostic résident dans la mémoire du PCM/CMM.

1.4.1.1. Exemples

Exemple 1 : Pour lire les entrées 9 à 16 de l'API Série 90 destinataire dans les entrées 17 à 24 de l'API Série 90 source, l'adresse source est 17, l'adresse destinataire est 9, et la longueur des données 8.

Exemple 2 : Pour lire l'entrée 27 de l'API Série 90 destinataire dans l'entrée 3 de l'API Série 90 source, vous devez spécifier une adresse source de 1, une adresse de destinataire de 25 et une longueur de données de 8. Les entrées 1 à 8 de la table des entrées de l'équipement source seront remplacées par les entrées 25 à 32 de la table des entrées de l'équipement destinataire.

1.4.2. Longueurs des données CCM

Le terme Longueur des données désigne la longueur du transfert de données. Les unités sont déterminées par le type de mémoire de l'équipement source et sont indiquées dans le tableau qui suit.

Tableau 5-6. Unités de longueur des types de mémoire CCM Série 90

Type de mémoire		Unité de longueur	Longueur possible
1 :	Registres	1 Registre =16 bits	Registre(s)
2,3 :	Entrées et sorties	1 Point = 1 bit	Multiple(s) de 8 Points
6 :	Zone de travail	1 Octet = 8 bits	Octet(s)
9 :	Mots d'état de diagnostic	1 Mot = 16 bits	Mot(s)
13,14 : 17,18 :	Entrées/Sorties Bit à 1 Entrées/Sorties Bit à 0	1 Point = 1 bit	Point

1.4.2.1. Exemples

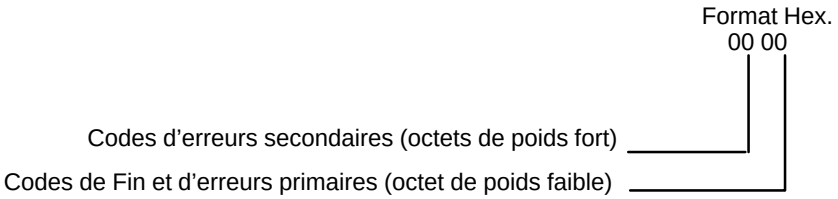
Exemple 1 : Pour lire 12 octets de la zone de travail de l'API Série 90 destinataire dans les registres Série 90, la longueur des données sera 6 puisque l'unité de longueur du type de mémoire de l'équipement source (registres) est le registre. Pour lire 12 mots d'état de diagnostic dans les registres, la longueur des données sera 12 puisque les registres et les mots d'état de diagnostic ont la même unité de longueur (registre=mot=2 octets).

Exemple 2 : Pour lire 8 entrées de l'API Série 90 destinataire dans les entrées de l'API Série 90, la longueur des données sera de 8 points puisque l'unité de longueur est la même pour les deux. Les types de mémoire CCM 2 et 3 (entrées et sorties) ne sont accessibles que par des multiples de 8.

Exemple 3 : Pour lire 8 registres de l'API Série 90 destinataire dans les entrées de l'API Série 90, la longueur des données sera de 8 registres multipliés par 16 points par registre = 128 points.

2. LE MOT D'ÉTAT CCM COMREQ

Le mot d'état CCM COMREQ rend compte de la progression et des résultats d'une demande CCM. L'adresse mémoire dans laquelle le CCM écrit le mot d'état est spécifiée dans les Mots 3 et 4 du bloc de commande. Le contenu du mot d'état CCM se définit comme suit :


Format Hex.
00 00
Codes d'erreurs secondaires (octets de poids fort)
Codes de Fin et d'erreurs primaires (octet de poids faible)

Lors de l'interprétation du mot d'état CCM COMREQ, plusieurs points sont à retenir :

1. Le protocole CCM n'enverra jamais un zéro pour le mot d'état CCM à la CPU de l'API. Si le programme utilisateur a besoin de savoir si la commande a fini de s'exécuter, il peut mettre le mot d'état CCM à zéro **avant** de générer l'instruction COMREQ et vérifier ensuite qu'il renferme une valeur non nulle.

2. Le code d'état 1 dans l'octet de poids faible indique que la demande a été satisfaite sans erreur. La liste complète des codes d'erreurs secondaires du CCM est fournie dans le tableau ci–après.

3. Affichez le mot d'état CCM sous le format hexadécimal pour pouvoir lire les deux octets de données. Si une erreur s'est produite, l'octet de poids faible sera supérieur à 1.

Le tableau ci–après indique les codes d'erreurs secondaires du CCM qui sont signalés (sous forme de codes d'erreurs secondaires) dans l'octet de poids fort du mot d'état CCM, après exécution d'un CCM COMREQ. Ces codes apparaissent également dans l'octet de poids faible du mot d'état de diagnostic 1 du CCM (DSW).

Tableau 5-7. Codes d'erreurs secondaires du CCM (Octet de poids fort du mot d'état COMREQ, Octet de poids faible de DSW)

Code d'erreur		Description
Décimal	**Hexadécimal**	
0	00	Transfert réussi.
1	01	Dépassement de temporisation sur la liaison série.
2	02	Un COMREQ a essayé d'écrire des données dans une section de la zone de travail CCM qui est protégée en écriture de façon permanente par le CCM.
3	03	Un COMREQ a essayé de lire ou d'écrire dans un point d'E/S inexistant.
4	04	Un COMREQ a essayé d'accéder à plus de données que n'en contient un type de mémoire particulier.
5	05	Un COMREQ a essayé de lire ou d'écrire un nombre d'octets non conforme dans la mémoire des registres ou dans les mots d'état de diagnostic.
6	06	Un COMREQ a essayé de lire ou d'écrire dans un ou plusieurs registres inexistants.
7	07	Un COMREQ a demandé le transfert de zéro octet de données.
8	08	Un COMREQ a essayé d'écrire dans une mémoire protégée.
9	09	Un COMREQ a essayé de transférer des données vers ou depuis un type de mémoire invalide ou l'adresse absolue d'un équipement source.
10	0A	Un COMREQ a essayé de lire ou d'écrire dans un ou plusieurs mots d'état de diagnostic inexistants.
11	0B	Un COMREQ a essayé de transférer des données sur une adresse invalide de la zone de travail ou sur une adresse de la table des entrées/sorties qui ne coïncide pas avec le début d'un octet (exemple, 1, 9, 17 ...).
12	0C	La communication série a été annulée après trois essais de transfert d'un bloc de données, ou après le nombre d'essais spécifié dans la configuration.
13	0D	La communication série a été annulée après trois essais de transfert d'un en–tête, ou après le nombre d'essais spécifié dans la configuration.
14	0E	La communication série a été annulée après trois essais de transfert d'une demande Q, ou après le nombre d'essais spécifié dans la configuration.
15	0F	Une tentative a été faite pour programmer les données d'une réponse Q sur un équipement qui n'est pas configuré comme un esclave.
20	14	Une ou plusieurs des erreurs suivantes se sont produites lors du transfert d'un bloc de données : a. Un caractère STX invalide a été reçu. b. Un caractère ETB invalide a été reçu. c. Un caractère ETX invalide a été reçu. d. Un caractère LRC invalide a été reçu. e. Une erreur de parité, de perte d'encadrement de trame ou de cadence s'est produite.

Tableau 5-7. Codes d'erreur secondaires du CCM (Octet de poids fort du mot d'état COMREQ, Octet de poids faible de DSW) (Suite)

Code d'erreur		Description
Décimal	**Hexadécimal**	
21	15	Le protocole CCM s'attendait à recevoir un caractère EOT d'un équipement externe et ne l'a pas reçu.
22	16	Le protocole CCM s'attendait à recevoir un caractère ACK ou NAK et n'a reçu ni l'un ni l'autre.
23	17	La communication a été annulée car le protocole CCM n'a reçu aucun accusé de réception valide en réponse à la séquence de demande d'un maître après 32 essais, ou le nombre d'essais spécifié dans la configuration.
24	18	La communication a été annulée après que la demande d'un équipement en mode Egal à égal ait reçu 32 accusés de réception négatifs (NACK) d'un équipement externe, ou le nombre spécifié dans la configuration.
25	19	La communication a été annulée car le protocole CCM n'a reçu aucune réponse valide à une demande en mode Egal à égal après 32 essais, ou le nombre d'essais spécifié dans la configuration.
26	1A	Un dépassement de temporisation s'est produit lors d'une tentative de transmission sur un port, le signal CTS étant resté trop longtemps dans l'état Inactif.
29	1D	Une erreur s'est produite lors du transfert des données entre le CCM et la CPU Série 90.
30	1E	Une erreur de parité, de perte d'encadrement de trame ou de cadence s'est produite lors du transfert d'un en–tête série.
31	1F	Une erreur de parité, de perte d'encadrement de trame ou de cadence s'est produite lors du transfert d'un bloc de données série.
34	22	Réception d'une réponse Q incorrecte.
48	30	Un COMREQ a essayé d'établir une conversation sur un port déjà en cours d'utilisation.
65	41	Le numéro de la commande COMREQ est invalide.
66	42	Une longueur de bloc de données COMREQ invalide a été spécifiée.
68	44	L'instruction COMREQ est invalide sur un port d'égal à égal.
69	45	L'instruction COMREQ est invalide sur un port esclave.
70	46	L'instruction COMREQ n'est valide que sur un port maître.
71	47	L'ID destinataire spécifié dans COMREQ est invalide.
72	48	Les valeurs spécifiées pour le bloc de données COMREQ sont hors gamme.
127	7F	Erreur diverse de type générique.

3. EXEMPLES DE PROGRAMMATION CCM COMREQ

Ce paragraphe décrit et donne un exemple de chaque commande CCM. Dans chaque cas, les valeurs à spécifier dans le bloc de commande sont indiquées. Un programme en diagramme en échelle est également fourni, à titre d'exemple.

Avant d'exécuter les diverses commandes CCM présentées ci–dessous, vous devez :

- Installer le module CMM dans le rack désiré (voir Chapitre 3), et connecter un câble entre Logicmaster 90 ou une miniconsole de programmation HHP et l'API.
- Configurer le rack et sélectionner les paramètres de communication désirés pour le module CMM. (Voir Chapitre 3). Vérifier que la configuration est valide.
- Mémoriser la configuration dans l'API. (Voir Chapitre 3).
- Installer le câble de communication série, si vous exécutez une commande déportée. (Voir les schémas de câbles dans le Chapitre 8).
- Ecrire et enregistrer le programme en diagramme en échelle ci–dessous ou un programme similaire.

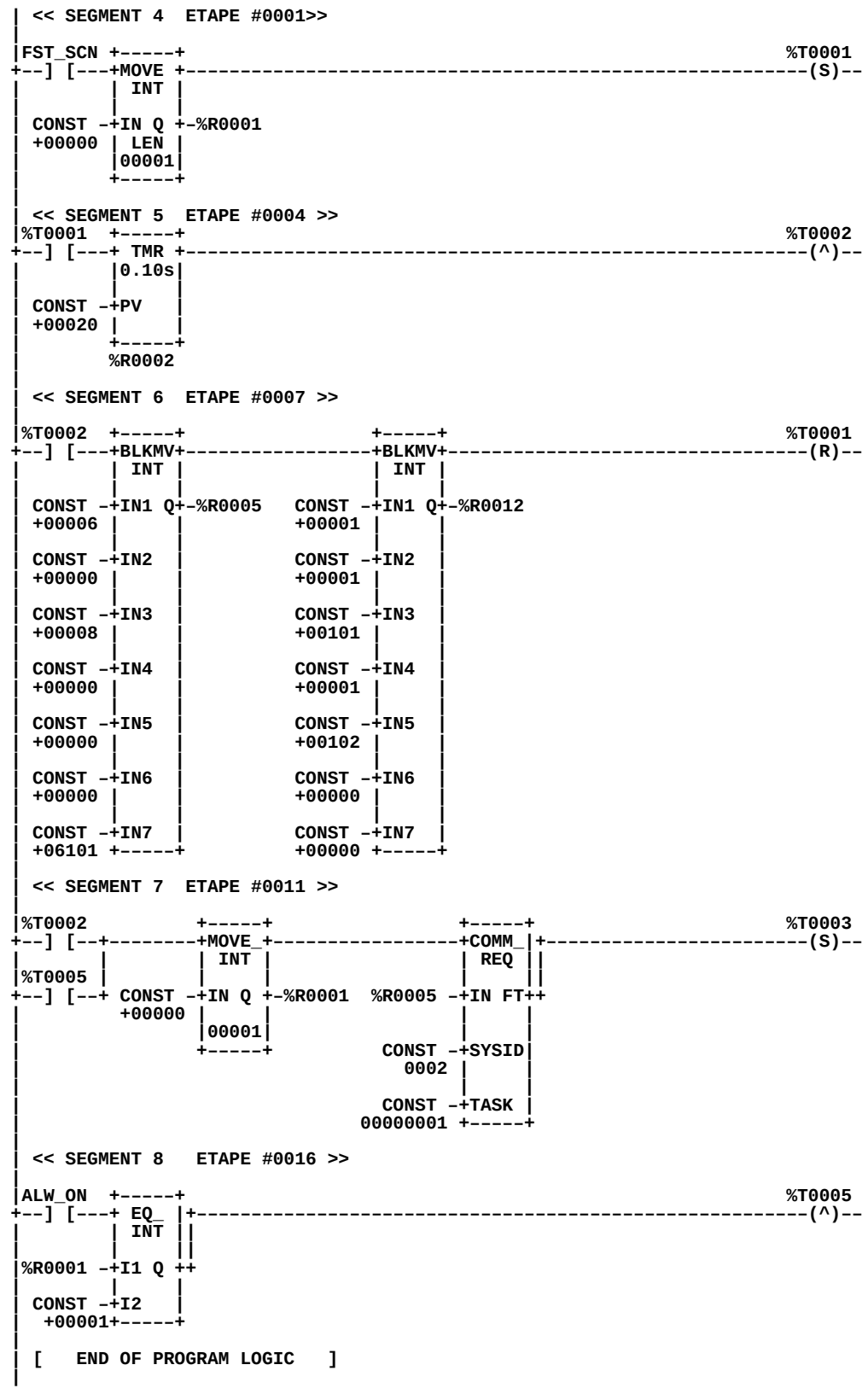
3.1. EXEMPLE DE PROGRAMME EN DIAGRAMME EN ÉCHELLE

Utilisez le programme en diagramme en échelle ci–après pour apprendre à programmer les commandes CCM et vérifier que votre système est correctement connecté et fonctionne normalement.

Pour les exemples qui suivent, remplacez simplement les valeurs du bloc de commande fourni dans les instructions BLOCK MOVE du programme en diagramme en échelle. Placez ensuite l'API en mode RUN. Vous pouvez afficher le mot d'état et les tables de données appropriées pour vérifier que la commande s'est exécutée correctement.

Dans le programme en diagramme en échelle fourni à titre d'exemple, les entrées de l'instruction COMREQ ont été définies comme suit :

- L'entrée IN désigne le Registre %R0005 comme début du bloc de commande.
- L'entrée SYSID indique que le module CMM associé est dans le rack 0, emplacement 2. Vérifiez que cela correspond à votre configuration système.
- L'entrée TASK indique que les commandes seront exécutées par le port 1 du module CMM.


| << SEGMENT 4 ETAPE #0001>>
|FST_SCN +-----+   %T0001
+--] [---+MOVE +---(S)--
| | INT |
| | |
| CONST -+IN Q +-%R0001
| +00000 | LEN |
| |00001|
| +-----+
| << SEGMENT 5 ETAPE #0004 >>
|%T0001 +-----+   %T0002
+--] [---+ TMR +---(^)--
| |0.10s|
| | |
| CONST -+PV |
| +00020 | |
| +-----+
| %R0002
| << SEGMENT 6 ETAPE #0007 >>
|%T0002 +-----+   +-----+   %T0001
+--] [---+BLKMV+----------------+BLKMV+---------------------------------(R)--
| | INT | | INT |
| | | | |
| CONST -+IN1 Q+-%R0005   CONST -+IN1 Q+-%R0012
+00006		+00001	
CONST -+IN2	CONST -+IN2
+00000		+00001	
CONST -+IN3	CONST -+IN3
+00008		+00101	
CONST -+IN4	CONST -+IN4
+00000		+00001	
CONST -+IN5	CONST -+IN5
+00000		+00102	
CONST -+IN6	CONST -+IN6
+00000		+00000	
CONST -+IN7	CONST -+IN7
+06101 +-----+   +00000 +-----+
<< SEGMENT 7 ETAPE #0011 >>
%T0002   +-----+   +-----+   %T0003
+--] [--+--------+MOVE_+----------------+COMM_|+----------------------(S)--
		INT		REQ	
%T0005					
+--] [--+ CONST -+IN Q +-%R0001 %R0005 -+IN FT++
| +00000 | | | |
| |00001| | |
| +-----+   CONST -+SYSID|
| 0002 | |
| | |
| CONST -+TASK |
| 00000001 +-----+
| << SEGMENT 8   ETAPE #0016 >>
|ALW_ON +-----+   %T0005
+--] [---+ EQ_ |+---(^)--
| | INT ||
| | ||
|%R0001 -+I1 Q ++
| | |
| CONST -+I2 |
| +00001+-----+
| [END OF PROGRAM LOGIC   ]

3.2. EXEMPLES DE COMMANDES CCM

Remarque

Dans chacun des exemples qui suit, le mode NOWAIT est sélectionné et le mot d'état est affecté au Registre 10 (%R00010).

3.2.1. Commande Programmer la réponse Q : 06001 (1771)

3.2.1.1. Description

Commande locale. La commande Programmer la réponse Q est destinée à un module CMM configuré en mode esclave seulement. Elle permet de transférer quatre octets de données sous le format Séquence Q de la CPU de l'API au module CMM. Un équipement maître déporté interrogera alors ce module CMM esclave à l'aide d'une commande Lire séquence Q (non disponible dans le module CMM) pour obtenir les données. La séquence Q élimine l'en–tête de 17 octets normalement inclus dans toutes les commandes de Lecture/Ecriture.

- Dans cette commande, les mots 8 et 9 du bloc de commande ont un rôle particulier :

Mot 8 :	Octets de données 1 et 2 de la Réponse Q
Mot 9 :	Octets de données 3 et 4 de la Réponse Q

- Format des octets de données :

	Octet de poids fort	**Octet de poids faible**
Mot 8 :	Octet de données 2	Octet de données 1
Mot 9 :	Octet de données 4	Octet de données 3

3.2.1.2. Exemple :

Programmer la réponse Q avec les nombres 1, 2, 3, 4.

Mot 1 :	00003 (0003)	Longueur du bloc de données
Mot 2 :	00000 (0000)	Mode NOWAIT
Mot 3 :	00008 (0008)	Type de mémoire du mot d'état (Registre)
Mot 4 :	00009 (0009)	Adresse du mot d'état moins 1 (Registre 10)
Mot 5 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 6 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 7 :	06001 (1771)	Numéro de la commande
Mot 8 :	00513 (0201)	Octets de données 1 et 2
Mot 9 :	01027 (0403)	Octets de données 3 et 4

3.2.2. Commande Effacer les mots d'état de diagnostic CCM : 06002 (1772)

3.2.2.1. Description

Commande locale. Cette commande n'a besoin que du numéro de la commande, Mot 7, pour s'exécuter.

3.2.2.2. Exemple

Effacer les mots d'état de diagnostic CCM

Mot 1 :	00001 (0001)	Longueur du bloc de données
Mot 2 :	00000 (0000)	Mode NOWAIT
Mot 3 :	00008 (0008)	Type de mémoire du mot d'état (Registre)
Mot 4 :	00009 (0009)	Adresse du mot d'état moins 1 (Registre 10)
Mot 5 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 6 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 7 :	06002 (1772)	Numéro de la commande

3.2.3. Commande Lire les mots d'état de diagnostic CCM dans les registres Source : 06003 (1773)

3.2.3.1. Description

Commande locale. 20 mots d'état de diagnostic CCM numérotés séquentiellement sont lisibles par la CPU de l'API. Tout ou partie de ces mots d'état de diagnostic peuvent être transférés dans la CPU, du moment qu'ils forment un bloc consécutif.

3.2.3.2. Exemple

Lire les cinq premiers mots d'état de diagnostic dans les registres %R00050–%R00054 de l'équipement source.

Mot 1 :	00006 (0006)	Longueur du bloc de données
Mot 2 :	00000 (0000)	Mode NOWAIT
Mot 3 :	00008 (0008)	Type de mémoire du Mot d'état (Registre)
Mot 4 :	00009 (0009)	Adresse de mot d'état moins 1 (Registre 10)
Mot 5 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 6 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 7 :	06003 (1773)	Numéro de la commande
Mot 8 :	00000 (0000)	Non utilisé
Mot 9 :	00000 (0000)	Non utilisé
Mot 10 :	00001 (0001)	Adresse mémoire destinataire
Mot 11 :	00005 (0005)	Longueur des données (mots)
Mot 12 :	00050 (0032)	Registre source

3.2.4. Commande de configuration logicielle : 06004 (1774)

3.2.4.1. Description :

Commande locale. Cette commande permet au programme d'application en diagramme en échelle de reconfigurer un port CCM avec des valeurs de temporisateurs et de compteurs d'essais spécifiques. Une fois générée, cette commande remplacera la configuration spécifiée à l'aide du logiciel Logicmaster 90. Toute valeur 65535 (FFFFh) attribuée à un paramètre indique que le choix courant du paramètre en question doit rester en vigueur. Les valeurs initiales des paramètres sont configurées à l'aide du logiciel Logicmaster 90.

3.2.4.2. Exemple :

Faire passer le temps de retournement à 1000 ms et le compteur d'essais ENQ à 16.

Mot 1 :	00015 (000F)	Longueur du bloc de données CCM
Mot 2 :	00000 (0000)	Mode NOWAIT
Mot 3 :	00008 (0008)	Type de mémoire du mot d'état (%R)
Mot 4 :	00009 (0009)	Adresse du mot d'état moins 1 (Registre 10)
Mot 5 :	00000 (0000)	Non utilisé
Mot 6 :	00000 (0000)	Non utilisé
Mot 7 :	06004 (1774)	Numéro de la commande
Mot 8 :	01000 (03E8)	Temps de retournement (0–65534 ms)
Mot 9 :	65535 (FFFF)	ENQ_ACK_TIMER (0–65534 ms)
Mot 10 :	65535 (FFFF)	SOH_TIMER (0–65534 ms)
Mot 11 :	65535 (FFFF)	HEADER_TIMER (0–65534 ms)
Mot 12 :	65535 (FFFF)	HEADER_ACK_TIMER (0–65534 ms)
Mot 13 :	65535 (FFFF)	STX_TIMER (0–65534 ms)
Mot 14 :	65535 (FFFF)	DATA_TIMER (0–65534 ms)
Mot 15 :	65535 (FFFF)	DATA_ACK_TIMER (0–65534 ms)
Mot 16 :	65535 (FFFF)	EOT_TIMER (0–65534 ms)
Mot 17 :	00016 (0010)	ENQ_COUNT (0–50)
Mot 18 :	65535 (FFFF)	HEADER_COUNT (0–50)
Mot 19 :	65535 (FFFF)	DATA_BLK COUNT (0–50)
Mot 20 :	65535 (FFFF)	Mode RS–485 (2, 4, ou FFFF)
Mot 21 :	65535 (FFFF)	Revalider le récepteur RS–485 (0–65534 ms)

Pour plus d'informations sur l'utilisation et la configuration de chaque temporisateur et compteur CCM, voir le Chapitre 7, Protocole CCM.

Le paramètre "Mode RS–485" permet de configurer le gestionnaire RS–485 pour qu'il fonctionne comme un émetteur 2 fils ou 4 fils. 4 fils est le mode de fonctionnement par défaut. Le choix 2 fils neutralise le récepteur RS–485 pendant que le gestionnaire RS–485 émet. Le récepteur RS–485 est revalidé dès que l'émetteur a fini et que le délai spécifié dans Revalider le récepteur a expiré.

Le paramètre "Revalider le récepteur" permet au gestionnaire RS–485 de retarder la revalidation des récepteurs RS–485 après que le gestionnaire ait fini d'émettre. Le retard doit être spécifié en millisecondes dans la gamme 0–65534 ms.

3.2.5. Commande Lire la mémoire de l'équipement destinataire dans la mémoire de l'équipement source : 06101–06103 (17D5–17D7)

3.2.5.1. Description

Commande déportée. Ce jeu de commandes permet de lire les informations contenues dans un équipement destinataire dans l'un des trois types de mémoire source mentionnés ci–dessous :

Type de mémoire source	Numéro de la commande
Table des registres	06101
Table des entrées	06102
Table des sorties	06103

Les types de mémoire destinataire suivants sont accessibles :

Type de mémoire destinataire	Numéro du type
Table des registres	1
Table des entrées	2
Table des sorties	3
Zone de travail CCM	6
Mots d'état de diagnostic	9

3.2.5.2. Exemple

Lire les mots d'état de diagnostic CCM 1–9 de l'équipement destinataire dans les registres %R00936–%R00944 de l'équipement source. L'ID CPU de l'équipement destinataire est 36.

Mot 1 :	00006 (0006)	Longueur du bloc de données
Mot 2 :	00000 (0000)	Mode NOWAIT
Mot 3 :	00008 (0008)	Type de mémoire du mot d'état (Registre)
Mot 4 :	00009 (0009)	Adresse du mot d'état moins 1 (Registre 10)
Mot 5 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 6 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 7 :	06101 (17D5)	Numéro de la commande
Mot 8 :	00036 (0024)	ID CPU de l'équipement destinataire
Mot 9 :	00009 (0009)	Type de mémoire de l'équipement destinataire
Mot 10 :	00001 (0001)	Adresse mémoire de l'équipement destinataire
Mot 11 :	00009 (0009)	Longueur des données
Mot 12 :	00936 (03A8)	Adresse mémoire de l'équipement source

Remarque

Si vous utilisez les tables des entrées et des sorties, l'adresse mémoire doit coïncider avec le début d'un octet et la longueur des données doit être un multiple de 8.

3.2.6. Commande Lire la réponse Q dans la table des registres de l'équipement source : 06109 (17DD)

3.2.6.1. Description

Commande déportée. Cette commande permet à la console de programmation de l'API de lire la mémoire–tampon de la Réponse Q d'un équipement déporté et d'enregistrer les données dans une adresse spécifique de la table des registres. La mémoire–tampon d'une Réponse Q renferme exactement 2 registres de données.

3.2.6.2. Exemple

Pour lire la Réponse Q de l'équipement CCM esclave 5 dans les registres %R00100–%R00101.

Mot 1 :	00006 (0006)	Longueur du bloc de données CCM
Mot 2 :	00000 (0000)	Mode NOWAIT
Mot 3 :	00008 (0008)	Type de mémoire du mot d'état (%R)
Mot 4 :	00009 (0009)	Adresse du mot d'état moins 1 (Registre 10)
Mot 5 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 6 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 7 :	06109 (17DD)	Numéro de la commande
Mot 8 :	00005 (0005)	ID CPU de l'équipement destinataire (ID esclave=5)
Mot 9 :	00000 (0000)	Non utilisé
Mot 10 :	00000 (0000)	Non utilisé
Mot 11 :	00000 (0000)	Non utilisé
Mot 12 :	00100 (0064)	Adresse mémoire de l'équipement source (Registre 100)

3.2.7. Commande Ecrire un seul bit : 06110 (17DE)

3.2.7.1. Description

Commande déportée. Cette commande permet à l'utilisateur de mettre à 1 ou à 0 un seul bit de la table des entrées ou des sorties d'une autre CPU.

Les types de mémoire destinataire/fonctions d'écriture d'un bit sont les suivants :

Type de mémoire destinataire	Numéro du type	Fonction binaire
Table des entrées	13	Bit à 1
Table des sorties	14	Bit à 1
Table des entrées	17	Bit à 0
Table des sorties	18	Bit à 0

3.2.7.2. Exemple

Mettre à 0 la sortie %Q00713 dans l'API Série 90 destinataire. L'ID CPU de l'équipement destinataire est 25.

Mot 1 :	00004 (0004)	Longueur du bloc de données
Mot 2 :	00000 (0000)	Mode NOWAIT
Mot 3 :	00008 (0008)	Type de mémoire du mot d'état (Registre)
Mot 4 :	00009 (0009)	Adresse du mot d'état moins 1 (Registre 10)
Mot 5 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 6 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 7 :	06110 (17DE)	Numéro de la commande
Mot 8 :	00025 (0019)	ID CPU de l'équipement destinataire
Mot 9 :	00018 (0012)	Type/Fonction de la mémoire de l'équipement destinataire
Mot 10 :	00713 (02C9)	Adresse mémoire de l'équipement destinataire

3.2.8. Commande Ecrire dans l'équipement destinataire depuis l'équipement source : 06111–06113 (17DF–17E1)

3.2.8.1. Description

Commande déportée. Ce jeu de commandes permet d'écrire des informations dans l'équipement destinataire à partir de l'un des trois types de mémoire mentionnés ci–dessous :

Type de mémoire source	Numéro de la commande
Table des registres	06111
Table des entrées	06112
Table des sorties	06113

Les types de mémoire destinataire dans lesquels on peut écrire sont les suivants :

Type de mémoire destinataire	Numéro du type
Table des registres	1
Table des entrées	2
Table des sorties	3

3.2.8.2. Exemple

Ecrire dans les registres %R00200–%R00249 de l'équipement destinataire le contenu des registres %R00001–%R0050 de l'équipement source. L'ID CPU de l'équipement destinataire est 10.

Mot 1 :	00006 (0006)	Longueur du bloc de données
Mot 2 :	00000 (0000)	Mode NOWAIT
Mot 3 :	00008 (0008)	Type de mémoire du mot d'état (Registre)
Mot 4 :	00009 (0009)	Adresse du mot d'état moins 1 (Registre 10)
Mot 5 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 6 :	00000 (0000)	Non utilisé en mode NOWAIT
Mot 7 :	06111 (17DF)	Numéro de la commande
Mot 8 :	00010 (000A)	ID CPU de l'équipement destinataire
Mot 9 :	00001 (0001)	Type de mémoire de l'équipement destinataire
Mot 10 :	00200 (00C8)	Adresse mémoire de l'équipement destinataire
Mot 11 :	00050 (0032)	Longueur des données
Mot 12 :	00001 (0001)	Adresse mémoire de l'équipement source

Remarque

Si vous utilisez les tables des entrées et des sorties, l'adresse mémoire doit coïncider avec le début d'un octet et la longueur des données doit être un multiple de 8.

Chapitre 6 Service SNP

Ce chapitre décrit le service de communication série SNP. Pour bien comprendre, il est important d'avoir lu le Chapitre 4, Etablissement de la communication – instruction COMREQ.

Ce chapitre traite des sujets suivants :

- 1 : Le bloc de données SNP COMREQ
- 2 : Le mot d'état SNP COMREQ
- 3 : Exemples de programmation SNP COMREQ

1. LE BLOC DE DONNÉES SNP COMREQ

Le bloc de données SNP COMREQ est la partie du bloc de commande COMREQ qui identifie la commande SNP à exécuter et fournit les paramètres de cette commande.

Ce paragraphe traite des sujets suivants :

- Structure du bloc de données SNP
- Types de mémoire SNP et Adressage

1.1. STRUCTURE DU BLOC DE DONNÉES SNP

A la différence du bloc de données CCM, le bloc de données SNP n'a pas un format unique. Ceci s'explique par la grande variété de tâches exécutées par les commandes SNP. Certaines commandes telles que Effacer les mots d'état de diagnostic ou Etat court de l'API ne requièrent qu'un nombre minimal de mots, voire un seul. D'autres, telles que Etablir un datagramme peuvent nécessiter beaucoup de mots pour spécifier tous les paramètres nécessaires à la commande.

1.2. TYPES DE MÉMOIRE SNP ET ADRESSAGE

La plupart des paramètres de l'instruction COMREQ exigent qu'un type de mémoire de l'API soit spécifié. Ces paramètres servent à identifier les zones de mémoire de l'équipement maître et de l'équipement esclave sur lesquelles porteront les opérations de lecture et d'écriture. Le tableau ci–dessous indique les types de mémoire que vous pouvez spécifier, les codes d'accès et les unités de longueur par lesquels chaque type de mémoire peut être accédé, ainsi que les gammes valides. Les gammes valides peuvent varier suivant le modèle de la CPU de l'API Série 90.

Tableau 6-1. Types de mémoire, unités de longueurs et gammes valides

Type de mémoire de l'API	Code du Type		Unité	Gamme valide(1)
	Déc	Hex	Longueur	
Registres (%R)	08	08h	mot	1–maximum
Entrées analogiques (%AI)	10	0Ah	mot	1–maximum
Sorties analogiques (%AQ)	12	0Ch	mot	1–maximum
Entrées logiques (%I)	70	46h	bit	1–maximum
	16	10h	octet	1–maximum(2)
Sorties logiques (%Q)	72	48h	bit	1–maximum
	18	12h	octet	1–maximum(2)
Temporaires logiques (%T)	74	4Ah	bit	1–maximum
	20	14h	octet	1–maximum (2)
Internes logiques (%M)	76	4Ch	bit	1–maximum
	22	16h	octet	1–maximum(2)
Logiques (%SA)	78	4Eh	bit	1–maximum
	24	18h	octet	1–maximum(2)
Logiques (%SB)	80	50h	bit	1–maximum
	26	1Ah	octet	1–maximum (2)
Logiques (%SC)	82	52h	bit	1–maximum
	28	1Ch	octet	1–maximum(2)
Logiques (%S) (lecture seule)	84	54h	bit	1–maximum
	30	1Eh	octet	1–maximum(2)
Données globales Genius (%G)	86	56h	bit	1–maximum
	56	38h	octet	1–maximum (2)

1 Les gammes maximales adressables pour chaque type de mémoire dépendent du modèle de CPU et de la configuration de la mémoire.

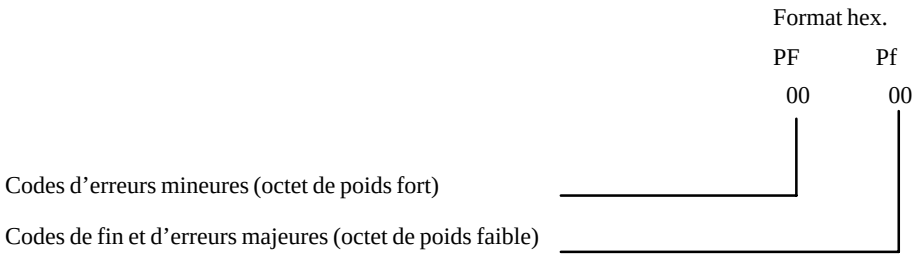
2 Avec un type de mémoire orienté Octet, les décalages d'adresse mémoire correspondants et le nombre d'éléments se spécifient en octets, et non pas en bits.

Remarque

Il n'y a pas de différence entre les types de mémoire orientés Bit et Octet en ce qui concerne la vitesse de traitement, la longueur des messages ou le temps de transfert des messages. Des restrictions de type de mémoire spéciales s'appliquent aux formats Point des datagrammes.

2. LE MOT D'ÉTAT SNP

Le mot d'état SNP COMREQ rend compte de la progression et des résultats d'une demande de communication SNP. L'adresse mémoire dans laquelle le CCM écrit le mot d'état est spécifiée dans les Mots 3 et 4 du bloc de commande. Le contenu du mot d'état SNP se définit comme suit :



Lors de l'interprétation du mot d'état SNP COMREQ, plusieurs points sont à retenir :

1. Le protocole CCM n'enverra jamais un zéro pour le mot d'état SNP à la CPU de l'API. Si le programme utilisateur a besoin de savoir si la commande a fini de s'exécuter, il peut mettre le mot d'état SNP à zéro **avant** de générer l'instruction COMREQ et vérifier ensuite qu'il renferme une valeur non nulle.
2. Le code d'état 1 dans l'octet de poids faible et 0 dans l'octet de poids fort indiquent que la demande a été satisfaite sans erreurs. Toutes les autres valeurs non nulles indiquent des erreurs. La liste complète des codes d'erreurs majeures et mineures du SNP est fournie dans les tableaux ci–après.
3. Affichez le mot d'état SNP sous le format hexadécimal pour pouvoir lire les deux octets de données.
4. Si une erreur s'est produite, l'octet de poids faible (code d'erreur majeure) sera supérieur à 1.
5. L'octet de poids fort contiendra tout code d'erreur mineure applicable.

Les tableaux ci–après indiquent les codes d'erreur SNP qui sont signalés dans le mot d'état SNP, après exécution d'un SNP COMREQ. Si une erreur s'est produite, ces codes apparaissent également dans le premier mot des mots d'état de diagnostic SNP (DSW 1).

2.1. CODES D'ERREURS MAJEURES SNP

Tableau 6-2. Codes d'erreurs majeures du SNP

Code d'erreur		Description des erreurs majeures
Décimal	**Hexadécimal**	
1	01h	Transfert réussi. (C'est la valeur normalement attendue dans le mot d'état COMREQ).
2	02h	Niveau de protection insuffisant. Pour l'API Série 90–70, le code d'erreur mineure renferme le niveau de protection requis pour la demande de service.
4	04h	Erreur de séquence du protocole. La CPU a reçu un message qui ne respecte pas l'ordre.
5	05h	Erreur de demande de service. Le code d'erreur mineure renferme le code d'erreur spécifique. Voir Tableau des codes d'erreurs mineures ci–après.
6	06h	Type de boîte aux lettres illégal. Le type de boîte aux lettres pour la demande de service est soit indéfini soit inattendu.
7	07h	La file d'attente Demande de service de la CPU de l'API est pleine. L'équipement maître doit refaire un essai plus tard. L'équipement maître doit attendre au minimum 10 ms avant d'envoyer une nouvelle demande de service.
10	0Ah	Erreur du gestionnaire SNP DOS. Le code d'erreur mineure renferme le code d'erreur spécifique. Voir Tableau des codes d'erreurs mineures ci–après.
11	0Bh	Demande de service illégale. Le service demandé est soit non défini soit non supporté. (Cette valeur est retournée au lieu de la valeur 01h qui est transmise dans le message d'erreur SNP, pour éviter tout risque de confusion avec le code de transfert COMREQ réussi normal).
12	0Ch	Erreur SNP/SNP–X locale. Une erreur s'est produite dans la tâche SNP du module CMM installé dans cet API. Cette erreur peut survenir soit dans le SNP maître, soit dans le SNP esclave. Le code d'erreur mineure renferme le code d'erreur spécifique. Voir Tableau des codes d'erreurs mineures ci–après.
13	0Dh	Erreur SNP déportée. Une erreur s'est produite dans la tâche SNP esclave du module CMM installé dans l'API déporté. Le code d'erreur mineure renferme le code d'erreur spécifique. Voir Tableau des codes d'erreurs mineures ci–après.
14	0Eh	Erreur de numérotation automatique. Une erreur s'est produite lors d'une tentative d'envoi d'une chaîne de commande à un modem externe. Le code d'erreur mineure renferme le code d'erreur spécifique. Voir Tableau des codes d'erreurs mineures ci–après.
15	0Fh	Erreur dans équipement SNP–X esclave. Une erreur s'est produite dans la tâche SNPX de l'équipement esclave déporté. Le code d'erreur mineure renferme le code d'erreur spécifique. Voir Tableau des codes d'erreurs mineures ci–après.
80	50h	Problème d'envoi de message à la tâche Demande de service de l'équipement esclave. (CPU des API Série 90–70 seulement).
81	51h	Problème de réception de message envoyé par la tâche Demande de service de l'équipement esclave. (CPU des API Série 90–70 seulement).
85	55h	Le temps limite de la tâche SNP dans l'équipement esclave a expiré avant réception d'une réponse à la demande de service. (CPU des API Série 90–70 seulement).
86	56h	La tâche SNP dans l'équipement esclave n'a pas trouvé la connexion de datagramme demandée. (CPU des API Série 90–70 seulement).
87	57h	La tâche SNP dans l'équipement esclave a détecté une erreur en essayant d'écrire le datagramme. (CPU des API Série 90–70 seulement).
88	58h	La tâche SNP dans l'équipement esclave a détecté une erreur en essayant de mettre à jour le datagramme. (CPU des API Série 90–70 seulement).

2.2. CODES D'ERREURS MINEURES SNP

La signification de chaque code d'erreur mineure dépend du code d'erreur majeure pour lequel il est défini. Consultez le tableau des codes d'erreurs mineures correspondant au code d'erreur majeure indiqué.

Tableau 6-3. Codes d'erreurs mineures correspondant au code d'erreur majeure 5

Code d'erreur		Description Erreur de demande de service
Décimal	**Hexadécimal**	
–1	FFh	La demande de service a été arrêtée prématurément.
–2	FEh	Pas de protection pour l'opération demandée.
–3	FDh	Il s'est avéré impossible de procéder à la configuration automatique.
–4	FCh	La configuration des E/S n'est pas valide.
–5	FBh	Impossible d'effacer la configuration des E/S.
–6	FAh	Impossible de remplacer le module d'E/S.
–7	F9h	Adresse de la tâche hors gamme.
–8	F8h	Nom de tâche spécifié invalide.
–9	F7h	Obligatoire pour entrer en communication avec une tâche du service.
–10	F6h	Cycle spécifié invalide.
–11	F5h	Mot de passe invalide.
–12	F4h	Paramètre d'entrée invalide dans la demande.
–13	F3h	Discordance dans la configuration des E/S.
–14	F2h	Programme invalide dans l'impossibilité d'entrer en communication.
–15	F1h	Demande uniquement valide à partir de la console de programmation.
–16	F0h	Demande uniquement valide en mode Stop.
–17	EFh	La console de programmation est déjà connectée.
–18	EEh	Impossible de retourner la taille des blocs.
–19	EDh	Détection d'une erreur sur le bus VME.
–20	ECh	Tâche impossible à créer.
–21	EBh	Tâche impossible à supprimer.
–22	EAh	Non connecté à la demande de service du procédé.
–23	E9h	Sélecteur de type de mémoire non valide dans ce contexte.
–24	E8h	Aucune mémoire utilisateur disponible pour l'affectation demandée.
–25	E7h	Configuration invalide.
–26	E6h	Le numéro de modèle de la CPU ne concorde pas.
–27	E5h	Zone des fichiers DOS non formatée.
–28	E4h	Type de mémoire inexistant pour ce sélecteur.
–29	E3h	Le numéro de révision de la CPU ne concorde pas.
–30	E2h	IOS n'a pas pu supprimer la configuration ou type spécifié erroné.
–31	E1h	Aucune configuration E/S à lire ou à supprimer.
–32	E0h	Le service en cours de traitement n'a pas pu se connecter.
–33	DFh	Adresse de connexion du datagramme invalide.
–34	DEh	Taille de connexion du datagramme invalide.
–35	DDh	L'ID de connexion du datagramme indiqué est introuvable.
–36	DCh	Adresse de connexion introuvable.

Tableau 6-3. Codes d'erreurs mineures correspondant au code d'erreur majeure 5 (Suite)

Code d'erreur		Description Erreur de demande de service
Décimal	**Hexadécimal**	
–37	DBh	Sélecteur de type de mémoire invalide dans le datagramme.
–38	DAh	Pointeur vers les données vide dans le sélecteur de type de mémoire.
–39	D9h	Type de transfert invalide pour ce sélecteur de type de mémoire.
–40	D8h	Longueur de point non autorisée.
–41	D7h	Type de datagramme spécifié invalide.
–42	D6h	Dépassement de mémoire affectée à la connexion du datagramme.
–43	D5h	Nom de bloc spécifié dans le datagramme invalide.
–44	D4h	Discordance du checksum (total de contrôle) de la configuration.
–45	D3h	Accès en lecture ou écriture du Module programme utilisateur UPM (User Program Module) qui excède la fin du bloc.
–46	D2h	Paramètre mode Ecriture invalide.
–47	D1h	La taille du paquet ou celle du programme ne correspond pas à l'entrée.
–48	D0h	Un ou plusieurs des modules API configurés ont un niveau de révision non supporté.
–49	CFh	Equipement spécifié non disponible dans le système (absent).
–50	CEh	L'équipement spécifié ne dispose pas de suffisamment de mémoire pour gérer la demande.
–51	CDh	Essai de lecture d'un équipement qui ne renfermait aucune donnée.
–52	CCh	Les données sauvegardées dans l'équipement ont été altérées et ne sont plus fiables.
–53	CBh	Une erreur de communication ou de vérification d'écriture s'est produite lors d'une commande Sauver ou Restaurer.
–54	CAh	L'équipement est protégé en écriture.
–55	C9h	Connexion avec mémoire–tampon non nulle obligatoire pour les commandes de bloc.
–56	C8h	Le/les mots de passe sont déjà activés et ne peuvent pas être forcés à l'état Inactif.
–57	C7h	Les mots de passe sont inactifs et ne peuvent être ni activés ni désactivés.
–58	C6h	Les tâches existent dans le programme de commande (CP) mais le demandeur n'est pas connecté au CP principal.
–59	C5h	Aucune configuration Rack/Emplacement au niveau Tâches à lire ou à supprimer.
–60	C4h	Vérifiez carte FA ou défaillance de l'EEPROM.
–61	C3h	La longueur du texte ne concorde pas avec le type de trafic.
–62	C2h	La touche OEM est NULL (inactive).
–63	C1h	Transition d'état du bloc invalide.

Tableau 6-4. Codes d'erreurs mineures correspondant au code d'erreur majeure 10 (0Ah)

Code d'erreur		Description Erreur du gestionnaire SNP DOS
Décimal	**Hexadécimal**	
–110	92h	Pas de communication SNP. La communication a été perdue ou aucune communication n'a été établie.
–111	91h	Problème de communication SNP. La transmission a été arrêtée prématurément après le nombre d'essais maximum pour cause d'erreurs série (exemple, erreurs de parité, de séquence ou de perte d'encadrement de trame).
–112	90h	Détection d'un SNP BCC erroné. La transmission a été arrêtée prématurément après le nombre d'essais maximum pour cause de Code de contrôle de bloc (BCC) erroné.
–113	8Fh	Message SNP hors séquence. Le type de message SNP reçu n'était pas celui attendu.
–114	8Eh	Le port série PC configuré pour le gestionnaire de l'équipement SNP maître n'est pas ouvert ; aucune communication n'est possible.
–115	8Dh	Mauvaise version DOS. DOS version 2.0, ou ultérieure, obligatoire pour le gestionnaire SNP DOS.

Remarque

Les codes d'erreurs mineures 1–31 correspondent à des erreurs non bloquantes ; la communication SNP ou SNP–X n'est pas interrompue.

Tableau 6-5. Codes d'erreurs mineures correspondant au code d'erreur majeure 12 (0Ch)

Code d'erreur		Description Erreur SNP/SNP–X locale
Décimal	**Hexadécimal**	
1	01h	COMREQ de type WAIT non autorisé. Type NOWAIT obligatoire.
2	02h	La commande COMREQ n'est pas supportée.
3	03h	La communication SNP n'est pas active. Une nouvelle communication SNP doit être établie en envoyant un COMREQ de connexion ou connexion longue.
4	04h	L'équipement SNP esclave n'a pas répondu au message de connexion envoyé par l'équipement maître.
5	05h	Impossible d'écrire le mot d'état SNP dans la mémoire de l'API local. Type ou adresse mémoire du mot d'état probablement invalide.
6	06h	Le type de mémoire de l'équipement maître n'est pas valide dans cet API.
7	07h	L'adresse ou la longueur de mémoire spécifiée pour l'équipement maître est nulle.
8	08h	Impossible de lire ou d'écrire dans les adresses mémoire de l'équipement maître spécifiées dans COMREQ. Erreur généralement provoquée par une adresse mémoire invalide pour cet API. L'échange du message SNP peut avoir eu lieu.
9	09h	La longueur des données spécifiée pour l'équipement maître excède la taille des données maximale du module CMM (2048 octets). Spécifiez une longueur de données plus petite. Si la longueur totale des données excède cette valeur maximale, utilisez plusieurs COMREQ.
10	0Ah	Le type de mémoire de l'équipement esclave n'est pas spécifié ou est invalide.
11	0Bh	L'adresse mémoire de l'équipement esclave n'est pas spécifiée ou est nulle.
12	0Ch	La longueur du bloc de données COMREQ est trop petite. (Lorsque la longueur COMREQ attendue est de 6 mots ou moins, une erreur de spécification de la longueur peut générer d'autres codes d'erreurs mineures 6–11).
13	0Dh	Mot de départ ou longueur du mot d'état de diagnostic (DSW) invalide.
14	0Eh	Taille maximale des données du message SNP invalide. Vous devez spécifier une valeur paire entre 42– 2048 octets.
15	0Fh	Niveau de protection invalide. Doit être défini entre 0 et 4 ou à –1.
16	10h	Sélecteur de table des défauts invalide. Doit être défini à 1 pour la table des défauts d'E/S ou à 2 pour la table des défauts automate.
17	11h	Indice de départ de la table des défauts invalide. Doit être défini entre 1–32 pour la table des défauts d'E/S ou 1–16 pour la table des défauts automate.
18	12h	Compteur de défauts invalide. Doit être défini entre 1–32 pour la table des défauts d'E/S ou 1–16 pour la table des défauts automate.
19	13h	Définir mode Date/Heure de l'API invalide. Doit être défini entre 1–4.
20	14h	Valeur Définir mode Date/Heure, heure ou jour de la semaine de l'API invalide.
21	15h	Impossible d'obtenir l'heure/la date de l'API maître à partir de la CPU de l'API.
22	16h	Type d'API esclave invalide. Doit être défini à 0 pour un API Série 90–70 ou à 1 pour un API Série 90–30 ou Série 90–20.
23	17h	Type de datagramme invalide. Doit être défini à 01h pour un datagramme normal ou 81h (129) pour un datagramme permanent.
24	18h	Format(s) Point de datagramme absent(s) ou trop nombreux. Doit être défini entre 1–32.
25	19h	Format Point du datagramme invalide.
26	1Ah	La taille de la zone de datagramme est trop petite pour contenir les données de tous les formats Point spécifiés.
27	1Bh	Nombre invalide de noms de programmes de commande. Doit être défini entre 1–8.

Tableau 6-5. Codes d'erreurs mineures correspondant au code d'erreur majeure 12 (0Ch) (Suite)

Code d'erreur		Description Erreur SNP/SNP–X locale
Décimal	**Hexadécimal**	
28	1Ch	La demande SNP–X excède la taille de données maximale (1000 octets). Vous devez spécifier une longueur de données plus petite. Utilisez plusieurs COMREQ si nécessaire.
29	1Dh	Type de session de communication SNP–X invalide. Doit être défini à 0 pour un seul équipement esclave, ou à 1 pour plusieurs équipements esclaves.
30	1Eh	ID SNP de destination illégal pour équipement SNP–X esclave. Doit comporter entre 0–7 caractères ASCII, plus un caractère de fin Nul (00h). L'ID SNP nul (huit octets de 00h) peut être utilisé pour spécifier tout équipement unique. L'ID SNP de diffusion (huit octets de FFh) peut être utilisé pour spécifier *tous* les équipements esclaves sur la liaison série.
31	1Fh	L'ID SNP de destination ne correspond pas au type de session SNP–X. L'ID SNP de diffusion n'est pas admis dans une session avec un seul équipement SNP–X esclave. L'ID SNP nul n'est pas admis dans un session avec plusieurs équipements SNP–X esclaves.

Remarque

Les codes d'erreurs mineures 32–118 indiquent des erreurs bloquantes ; la prochaine demande de communication doit être initialisée avec un COMREQ de connexion ou connexion longue, ou un COMREQ SNP–X déporté.

Tableau 6-5. Codes d'erreurs mineures ccorrespondant au code d'erreur majeure 12 (0Ch) (Suite)

Code d'erreur		Description Erreur SNP/SNP–X locale
Décimal	**Hexadécimal**	
32	20h	Temps limite d'inactivité (T3'). L'équipement SNP esclave n'a reçu aucun nouveau message SNP dans l'intervalle de temps spécifié pour T3'.
33	21h	Une erreur de parité s'est produite dans un message de connexion, une réponse de connexion ou une commande Mettre le datagramme à jour en temps réel. La communication n'a pas été établie.
34	22h	Une erreur BCC (Code de Contrôle de bloc) s'est produite dans un message de connexion, une réponse de connexion ou une commande Mettre le datagramme à jour en temps réel. La communication n'a pas été établie.
35	23h	Une erreur série de perte d'encadrement de trame ou de séquence s'est produite dans un message de connexion, une réponse de connexion ou une commande Mettre le datagramme à jour en temps réel. La communication n'a pas été établie.
36	24h	Un type de message SNP invalide a été reçu à la place d'un message de connexion, une réponse de connexion ou une commande Mettre le datagramme à jour en temps réel. La communication n'a pas été établie.
37	25h	Une valeur "longueur du message suivant" invalide a été spécifiée dans un message de connexion, une réponse de connexion ou une commande Mettre le datagramme à jour en temps réel. La communication n'a pas été établie.
38	26h	Un type de message SNP inattendu a été reçu à la place d'un message de connexion, une réponse de connexion ou une commande Mettre le datagramme à jour en temps réel. La communication n'a pas été établie.
39	27h	Un autre Break a été reçu pendant qu'un équipement SNP esclave attendait un message de connexion, une réponse de connexion ou une commande Mettre à jour le datagramme en temps réel.
40	28h	Un message SNP a été envoyé et tenté d'être renvoyé le nombre maximum de fois. Deux essais maximum sont admis. Un essai de renvoi est tenté à réception d'un NAK provenant de l'équipement SNP déporté.

Tableau 6-5. Codes d'erreurs mineures correspondant au code d'erreur majeure 12 (0Ch) (Suite)

Code d'erreur		Description Erreur SNP/SNP–X locale
Décimal	**Hexadécimal**	
41	29h	Un message SNP reçu n'a pas été acquitté (NAK) le nombre maximum de fois. Le message non acquitté (NAK) peut être retransmis deux fois au maximum.
42	2Ah	Un message inconnu a été reçu à la place d'un message d'accusé de réception positif ou négatif (ACK ou NAK).
43	2Bh	Erreur de séquence. Un type de message SNP inattendu a été reçu.
44	2Ch	Un message SNP reçu renferme une valeur "longueur du message suivant" erronée.
45	2Dh	Dépassement du temps limite d'acquittement. Aucun accusé de réception positif ou négatif (ACK ou NAK) n'a été reçu dans l'intervalle de temps spécifié pour T2. Un équipement esclave peut générer cette erreur si l'équipement maître s'est arrêté après le nombre maximum de NAK et n'a pas accusé réception de l'essai suivant.
46	2Eh	Dépassement du temps limite de réponse. L'équipement SNP maître n'a pas reçu de message Réponse SNP dans l'intervalle de temps spécifié pour T5'.
47	2Fh	Dépassement de temps limite d'un message Tampon. Aucun message Tampon Texte ou Données de connexion n'a été reçu dans l'intervalle de temps spécifié pour T5".
48	30h	Dépassement de temps limite de la sortie série. Le module CMM n'a pas pu transmettre de Break, de message SNP, ou d'accusé de réception positif ou négatif SNP (ACK ou NAK) à partir du port série. (Peut provenir d'un signal CTS absent alors que le module CMM est configuré pour utiliser le contrôle de flux Matériel).
49	31h	L'équipement SNP esclave n'a pas reçu de réponse du processeur de demandes de services de la CPU de l'API.
50	32h	Dépassement du temps limite COMREQ. L'instruction COMREQ n'a pas fini de s'exécuter dans l'intervalle de temps spécifié.
51	33h	Une Demande ou une Réponse SNP a été arrêtée avant d'aboutir pour cause de réception d'un Break.
52	34h	Erreur de communication du fond de bac de l'API.
53	35h	Type ou adresse mémoire des données Etat complémentaire invalide. La communication n'a pas été établie.
54	36h	ID SNP de l'équipement SNP esclave invalide. Doit comporter entre 0–7 caractères ASCII, plus un caractère de fin Nul (00h). L'ID SNP nul (huit octets de 00h) peut être utilisé pour spécifier tout équipement unique.
55	37h	L'équipement SNP maître a reçu un message de réponse renfermant une longueur de données inattendue. Indique généralement un problème avec l'équipement SNP esclave déporté. Peut se produire lorsque des commandes Série 90–70 (Lecture/Ecriture de la mémoire des tâches ou de la mémoire des blocs du programme) sont générées sur un équipement Série 90–30 esclave.
56	38h	Le code de la réponse figurant dans la Réponse SNP–X reçue ne correspond pas à la valeur attendue. (Le code de la réponse doit être égal au code de la demande +80h).
57	39h	Le message de réponse SNP–X excède la taille de données maximale (1000 octets en décimal). Les données de la réponse sont ignorées.

Tableau 6-5. Codes d'erreurs mineures correspondant au code d'erreur majeure 12 (0Ch) (Suite)

Code d'erreur		Description Erreur SNP/SNP–X locale
Décimal	**Hexadécimal**	
64	40h	Une erreur de parité s'est produite dans un message Réponse de connexion X lors de l'établissement d'une nouvelle session de communication SNP–X. La communication n'a pas été établie.
65	41h	Une erreur de perte d'encadrement de trame ou de séquence s'est produite dans un message Réponse de connexion X lors de l'établissement d'une nouvelle session de communication SNP–X. La communication n'a pas été établie.
66	42h	Une erreur BCC (Code de contrôle de bloc) s'est produite dans un message Réponse de connexion X lors de l'établissement d'une nouvelle session de communication SNP–X. La communication n'a pas été établie.
67	43h	Un type de message invalide a été reçu à la place d'une Réponse de connexion X lors de l'établissement d'une nouvelle session de communication SNP–X. La communication n'a pas été établie.
68	44h	Une valeur "type du message suivant" invalide a été détectée dans un message Réponse de connexion X lors de l'établissement d'une nouvelle session de communication SNP–X. La communication n'a pas été établie.
69	45h	Un code de réponse invalide a été détecté dans un message Réponse de connexion X lors de l'établissement d'une nouvelle session de communication SNP–X. La communication n'a pas été établie.
70	46h	Le message Réponse de connexion X attendu n'a pas été reçu dans le temps limite défini pour la réponse, lors de l'établissement d'une nouvelle session de communication SNP–X. L'équipement maître a essayé de retransmettre deux fois le message de Connexion X, sans recevoir de réponse. La communication n'a pas été établie.
80	50h	Une erreur de parité s'est produite dans un message Réponse de connexion X lors du rétablissement d'une session de communication SNP–X existante. La communication n'a pas été établie.
81	51h	Une erreur de perte d'encadrement de trame ou de séquence s'est produite dans un message Réponse de connexion X lors du rétablissement d'une session de communication SNP–X existante. La communication n'a pas été établie.
82	52h	Une erreur BCC (Code de contrôle de bloc) s'est produite dans un message Réponse de connexion X lors du rétablissement d'une session de communication SNP–X existante. La communication n'a pas été établie.
83	53h	Un type de message invalide a été reçu à la place d'une Réponse de connexion X lors du rétablissement d'une session de communication SNP–X existante. La communication n'a pas été établie.
84	54h	Une valeur "type du message suivant" invalide a été détectée dans un message Réponse de connexion X lors du rétablissement d'une session de communication SNP–X existante. La communication n'a pas été établie.
85	55h	Un code de réponse invalide a été détecté dans un message Réponse de connexion X lors du rétablissement d'une session de communication SNP–X existante. La communication n'a pas été établie.
86	56h	Le message Réponse de connexion X attendu n'a pas été reçu dans le temps limite défini pour la réponse, lors du rétablissement d'une session de communication SNP–X existante. L'équipement maître a essayé de retransmettre deux fois le message de Connexion X, sans recevoir de réponse. La communication n'a pas été établie.

Tableau 6-5. Codes d'erreurs mineures correspondant au code d'erreur majeure 12 (0Ch) (Suite)

Code d'erreur		Description Erreur SNP/SNP–X locale
Décimal	**Hexadécimal**	
96	60h	Une erreur de parité s'est produite dans un message Réponse de connexion X.
97	61h	Une erreur de perte d'encadrement de trame ou de séquence s'est produite dans un message Réponse de connexion X.
98	62h	Une erreur BCC (Code de contrôle de bloc) s'est produite dans un message Réponse de connexion X.
99	63h	Un type de message invalide a été reçu à la place d'une Réponse de connexion X.
100	64h	Une valeur "type du message suivant" invalide a été détectée dans un message Réponse de connexion X.
101	65h	Un code de réponse invalide a été détecté dans un message Réponse de connexion X.
102	66h	Le message Réponse de connexion X attendu n'a pas été reçu dans le temps limite défini pour la réponse.
112	70h	Une erreur de parité s'est produite dans un message Réponse intermédiaire.
113	71h	Une erreur de perte d'encadrement de trame ou de séquence s'est produite dans un message Réponse intermédiaire.
114	72h	Une erreur BCC (Code de contrôle de bloc) s'est produite dans un message Réponse intermédiaire.
115	73h	Un type de message invalide a été reçu à la place d'une Réponse intermédiaire.
116	74h	Une valeur "type du message suivant" invalide a été détectée dans un message Réponse intermédiaire.
117	75h	Un code de réponse invalide a été détecté dans un message Réponse intermédiaire.
118	76h	Le message Réponse intermédiaire attendu n'a pas été reçu dans le temps limite défini pour la réponse.

Tableau 6-6. Codes d'erreurs mineures correspondant au code d'erreur majeure 13 (0Dh)

Code d'erreur		Description Erreur SNP déportée
Décimal	**Hexadécimal**	
64	40h	Le service demandé n'est pas supporté par l'équipement SNP esclave.
65	41h	L'équipement SNP esclave sur le module CMM nécessite le niveau de protection 2 de la CPU de l'API pour pouvoir fonctionner. L'équipement SNP esclave a rejeté une demande de modification du niveau de protection.
66	42h	Le message de demande ou de réponse SNP excède la longueur de données maximale autorisée dans le module CMM. (2048 octets pour les messages Boîtes aux lettres et tous les messages Tampon qui suivent). La longueur des données doit être réduite dans l'équipement maître. Utilisez si nécessaire plusieurs demandes.
67	43h	Format du message Ecrire le datagramme incorrect. Les équipements Série 90–70 esclaves utilisent pour ce message un format différent de celui utilisé par les équipements Série 90–30 ou Série 90–20 esclaves. L'équipement maître doit utiliser le format de message correspondant à ce type d'équipement SNP esclave. (L'équipement SNP maître dans le module CMM inclut ce message dans la commande COMREQ Etablir un datagramme. Le datagramme a été partiellement établi mais est inutilisable ; il est conseillé de l'annuler en utilisant l'ID retourné par la commande COMREQ).
68	44	Une erreur de datagramme s'est produite dans un équipement Série 90–70 esclave (erreur sur double port).

Tableau 6-7. Codes d'erreurs mineures correspondant au code d'erreur majeure 14 (0Eh)

Code d'erreur		Description Erreur Numérotation automatique
Décimal	**Hexadécimal**	
1	01h	Non utilisé.
2	02h	La longueur de la chaîne de commande du modem excède 250 caractères.
3	03h	La longueur du bloc de données COMREQ spécifiée est trop petite. Les données de la chaîne de commande sont absentes ou incomplètes.
4	04h	Dépassement du temps limite sur la sortie série. Le module CMM n'a pas réussi à transmettre la commande de numérotation automatique à partir du port série. (Peut provenir d'un signal CTS absent alors que le module CMM est configuré pour utiliser le contrôle de flux Matériel).
5	05h	Aucune réponse n'a été reçue du modem. Vérifiez le modem et le câble.
6	06h	Le modem a répondu par "BUSY" (occupé). Le modem n'a pas pu satisfaire la connexion demandée. Le modem déporté est déjà en cours d'utilisation ; essayez de redemander la connexion un peu plus tard.
7	07h	Le modem a répondu par "NO CARRIER" (pas de porteuse). Le modem n'a pas pu satisfaire la connexion demandée. Vérifiez les modems locaux et déportés ainsi que la ligne téléphonique.
8	08h	Le modem a répondu par "NO DIALTONE" (pas de tonalité). Le modem n'a pas pu satisfaire la connexion demandée. Vérifiez les connexions du modem ainsi que la ligne téléphonique.
9	09h	Le modem a répondu par "ERROR" (erreur). Le modem n'a pas pu satisfaire la connexion demandée. Vérifiez la chaîne de commande du modem et le modem.
10	0Ah	Le modem a répondu par "RING", ce qui signifie qu'il est actuellement appelé par un autre modem. Le modem n'a pas pu satisfaire la connexion demandée. Essayez de redemander la connexion un peu plus tard.
11	0Bh	Une réponse inconnue a été reçue du modem. Le modem n'a pas pu exécuter la commande demandée. Vérifiez la chaîne de commande du modem et le modem. Le modem doit normalement répondre par "CONNECT" ou "OK".

Remarque

Les codes d'erreurs mineures 1–21 indiquent des erreurs non bloquantes ; la session de communication SNP–X n'est pas interrompue.

Tableau 6-8. Codes d'erreurs mineures correspondant au code d'erreur majeure 15 (0Fh)

Code d'erreur		Description Erreur dans un équipement SNP–X esclave
Décimal	**Hexadécimal**	
1	01h	Le code de demande de service figurant dans le message Demande X n'est pas supporté ou est invalide à ce stade. Cette erreur peut se produire lorsqu'une session de communication SNP–X n'a pas pu être établie avec l'équipement esclave.
2	02h	Niveau de protection insuffisant dans la CPU de l'API esclave pour le service SNP–X demandé. Le mot de passe défini au niveau de la CPU de l'API peut empêcher le service demandé d'aboutir.
3	03h	Type de mémoire de l'équipement esclave invalide dans le message Demande X.
4	04h	Adresse ou gamme d'adresses mémoire invalide dans le message Demande X.
5	05h	Longueur des données invalide dans le message Demande X. La valeur spécifiée doit être différente de zéro et ne pas dépasser 1000 octets en décimal.
6	06h	La longueur des données Tampon–X ne correspond pas à la demande de service formulée. La longueur du message Tampon–X est calculée à partir de la valeur spécifiée dans le champ "Longueur du message suivant" du message Demande X ; la longueur des données dans le message Tampon est toujours égale à la longueur du message moins 8 octets.
7	07h	Indication "File d'attente pleine" de la part du processeur de demandes de service de la CPU de l'API esclave. L'équipement esclave est momentanément dans l'incapacité de satisfaire la demande de service ; l'équipement maître devra tenter un nouvel essai. Il est recommandé que l'équipement maître attende au moins 10 ms avant de réitérer la Demande X. (Cette erreur ne s'applique qu'au module CMM).
8	08h	La réponse du processeur de demandes de service excède 1000 octets ; l'équipement SNP–X esclave ne peut pas retourner les données dans un message Réponse X. (Cette erreur ne s'applique qu'au module CMM).
16	10h	Erreur inattendue du processeur de demandes de service. (Cette erreur ne s'applique qu'au module CMM ; le code de l'erreur SRP inattendue est sauvegardé dans les mots d'état de diagnostic du module CMM).
21	15h	Demande de service non autorisée dans une demande de diffusion. L'équipement maître doit adresser le message Demande X à un équipement SNP–X esclave spécifique.

Remarque

Les codes d'erreurs mineures 32–35 indiquent des erreurs bloquantes ; une nouvelle session de communication SNP–X doit être établie via un message de connexion X. L'équipement SNP–X esclave retourne ces codes d'erreur dans un message Réponse X.

Tableau 6-8. Codes d'erreurs mineures correspondant au code d'erreur majeure 15 (0Fh) (Suite)

Code d'erreur		Description Erreur dans un équipement SNP–X esclave
Décimal	**Hexadécimal**	
32	20h	Champ "Type de message" invalide dans un message Demande X reçu. Le type de message d'une Demande X doit être défini à 58h = 'X'.
33	21h	Champ "Type du message suivant" ou "Longueur du message suivant" invalide dans un message Demande X reçu. Si la demande n'utilise pas de tampon (0–2 octets de données), le Type du message suivant doit être défini à zéro. Si la demande est suivie d'un message Tampon (plus de 2 octets de données), le Type du message suivant doit être défini à 54h = 'T', et la Longueur du message suivant doit spécifier la longueur du message Tampon X. Vous pouvez spécifier une longueur comprise entre 9 et 1008 octets (longueur des données plus 8 octets).
34	22h	Champ "Type de message" invalide dans un message Tampon X reçu. Le type de message d'un Tampon X doit être défini à 54h = 'T'.
35	23h	Champ "Type du message suivant" invalide dans un message Tampon X reçu. Etant donné qu'un message Tampon X n'est jamais suivi d'un autre message, le Type du message suivant doit toujours être défini à zéro.

Remarque

Les codes d'erreurs mineures 64–115 indiquent des erreurs bloquantes ; une nouvelle session de communication SNP–X doit être établie via un message de connexion X. L'équipement SNP–X esclave ne peut pas retourner de Réponse X ; ces codes d'erreur ne sont accessibles que par les mots d'état de diagnostic mis régulièrement à jour pour chaque port série du module CMM.

Tableau 6-8. Codes d'erreurs mineures correspondant au code d'erreur majeure 15 (0Fh) (Suite)

Code d'erreur		Description Erreur dans un équipement SNP–X esclave
Décimal	**Hexadécimal**	
64	40h	Dépassement du temps limite sur la sortie série. Le module CMM n'a pas réussi à transmettre un message SNP–X à partir du port série. (Peut provenir d'un signal CTS absent alors que le module CMM est configuré pour utiliser le contrôle de flux Matériel).
65	41h	Une demande SNP–X a été arrêtée avant la fin pour cause de réception d'un Break.
66	42h	Réception d'un message Tampon X contenant plus de 1000 octets de données. Les données sont ignorées.
67	43h	L'équipement SNP–X esclave n'a reçu aucune réponse du processeur de demandes de service résidant dans la CPU de l'API.
68	44h	Erreur de communication dans fond de bac de l'API.
80	50h	Une erreur de parité s'est produite dans un message Connexion X reçu.
81	51h	Une erreur de perte d'encadrement de trame ou de séquence s'est produite dans un message Connexion X reçu.
82	52h	Une erreur BCC (Code de contrôle de bloc) s'est produite dans un message Connexion X reçu.
83	53h	Un type de message invalide a été reçu à la place d'un message Connexion X. (Pour un message Connexion X, le type du message doit être défini à 58h = 'T').
84	54h	Une valeur "Type du message suivant" invalide a été détectée dans un message Connexion X reçu. (Pour un message Connexion X, la Longueur du message suivant doit être définie à zéro).
85	55h	Un code de demande invalide a été détecté dans un message Connexion X reçu.
96	60h	Une erreur de parité s'est produite dans un message Demande X reçu.
97	61h	Une erreur de perte d'encadrement de trame ou de séquence s'est produite dans un message Demande X reçu.
98	62h	Une erreur BCC (Code de contrôle de bloc) s'est produite dans un message Demande X reçu.
112	70h	Une erreur de parité s'est produite dans un message Tampon X reçu.
113	71h	Une erreur de perte d'encadrement de trame ou de séquence s'est produite dans un message Tampon X reçu.
114	72h	Une erreur BCC (Code de contrôle de bloc) s'est produite dans un message Tampon X reçu.
115	73h	Le message Tampon X attendu n'a pas été reçu.

3. EXEMPLES DE PROGRAMMATION DE SNP COMREQ

Ce paragraphe décrit et donne un exemple de chaque commande SNP. Dans chaque cas, les valeurs à spécifier dans le bloc de commande sont indiquées. Un programme en diagramme en échelle est également fourni, à titre d'exemple.

Avant d'exécuter les diverses commandes SNP présentées ci–dessous, vous devez :

- Installer le module CMM dans le rack désiré (voir Chapitre 3).
- Configurer le rack et sélectionner les paramètres de communication désirés pour le module CMM, à l'aide du logiciel de configuration Logicmaster 90 ou de la miniconsole de programmation HHP (voir Chapitre 3).
- Vérifier que la configuration du rack est valide.
- Mémoriser la configuration du rack dans l'API (voir Chapitre 3). Si vous utilisez le logiciel Logicmaster 90, avoir vérifié que la configuration de l'API correspond à la configuration Logicmaster 90.
- Installer le câble de communication série entre l'équipement maître et les équipements esclaves pour toute commande déportée (voir les schémas de câblage dans le Chapitre 8).
- Ecrire et enregistrer le programme en diagramme en échelle ci–dessous ou un programme similaire dans l'API.

Remarque

Dans les exemples qui suivent, le module CMM a été installé et configuré dans le rack 0, emplacement 2 et le protocole SNP a été validé sur le Port 1. Si votre configuration diffère de celle–ci, modifiez les champs COMREQ correspondant au rack/à l'emplacement (champ SYSID) et /ou le numéro du port (champ TACHE) pour qu'ils reflètent votre configuration.

3.1. EXEMPLE DE PROGRAMME EN DIAGRAMME EN ÉCHELLE

Utilisez le programme en diagramme en échelle ci–après pour apprendre à programmer les commandes CCM et vérifier que votre système est correctement connecté et fonctionne normalement.

Le programme en diagramme en échelle ci–dessous établit une communication SNP point–à–point avec tout équipement SNP esclave en générant une commande de connexion unique peu après le démarrage du programme et en générant de façon répétitive des commandes Lire Mémoire système sur l'équipement esclave connecté.

Pour utiliser les autres commandes figurant dans le programme en diagramme en échelle fourni, remplacez simplement les valeurs du bloc de commande fournies pour chaque commande dans les instructions BLOCK MOVE des segments 6 ou 9 du programme. (Le segment 6 prépare la commande de connexion initiale ; le segment 9 prépare la commande répétitive qui suit la demande de connexion initiale). Le programme proposé charge jusqu'à 8 mots dans le bloc de données COMREQ ; ajoutez des instructions BLOCK MOVE dans le programme si la commande que vous désirez exécuter nécessite davantage de données.

Dans l'exemple, les deux premiers segments (**Segment 4** et **Segment 5**) fournissent une temporisation de 2 secondes au démarrage du programme. Cela laisse au module CMM plus de temps que nécessaire pour s'initialiser au démarrage du système. Une fois les 2 secondes écoulées, le temporisateur monocoup %T0002 se déclenche pour charger la commande de connexion.

Le Segment 6 charge une commande de connexion (07200) dans le bloc de commande COMREQ, après la temporisation de la mise sous tension. Cette commande utilise l'ID SNP nul pour pouvoir communiquer avec n'importe quel équipement esclave. (La signalisation de l'état complémentaire n'est pas validée).

Le Segment 7 active l'instruction COMREQ pour envoyer la commande SNP au module CMM. Le bloc de commande a préalablement été configuré dans les registres %R0005 et suivants. Le mot d'état SNP dans %R0001 est effacé ; il sera mis à jour par le module CMM une fois que la commande aura fini de s'exécuter.

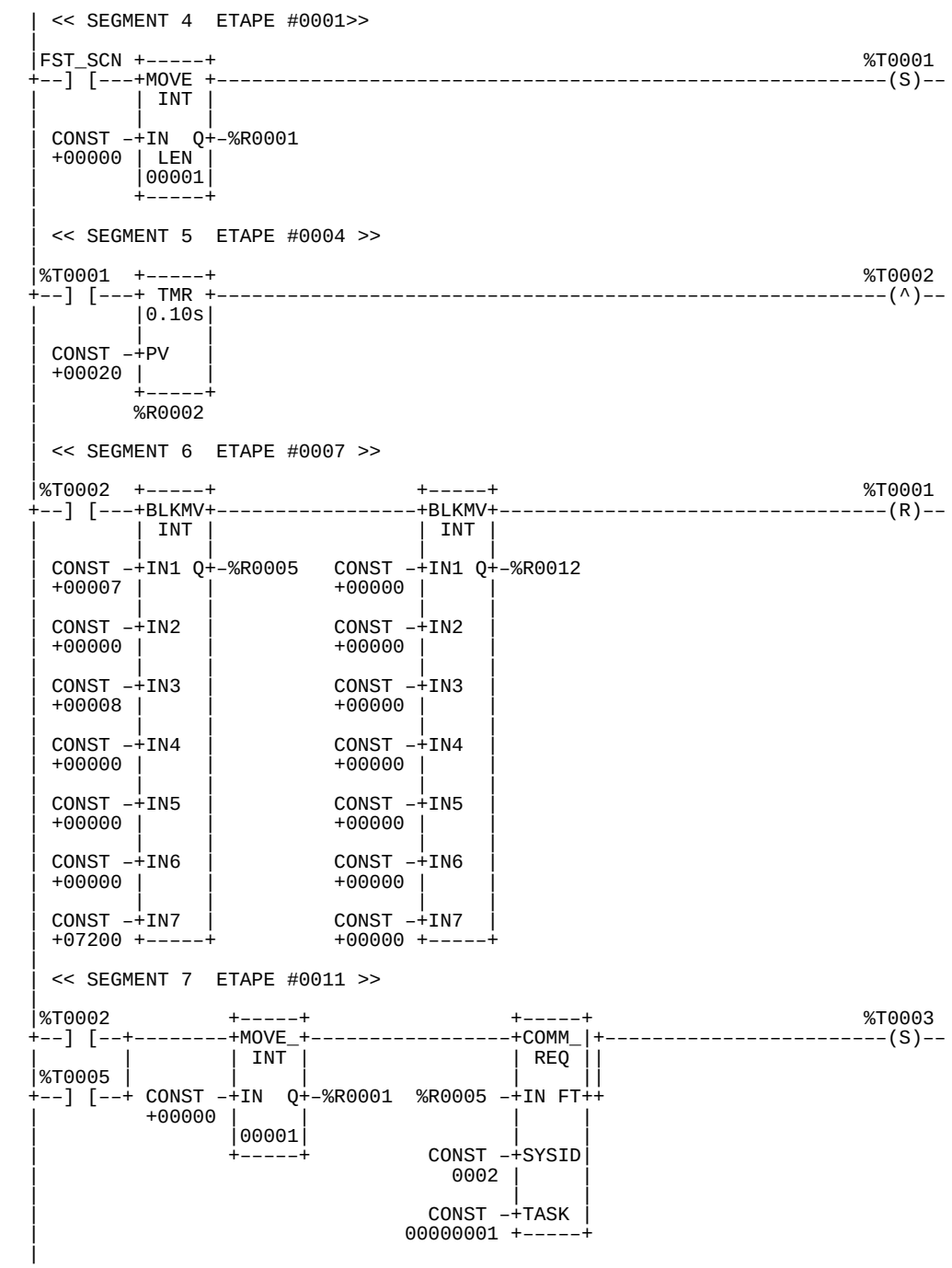
Le Segment 8 surveille le mot d'état SNP. Lorsqu'il indique 0001 (Transfert réussi), le temporisateur %T0004 se déclenche pour charger la commande SNP répétitive.

Le Segment 9 charge une commande Lire Mémoire système (07202) dans le bloc de commande dès que la précédente commande a fini de s'exécuter. Cette commande lit le registre %R101 de l'équipement esclave dans le registre %R102 de l'équipement maître.

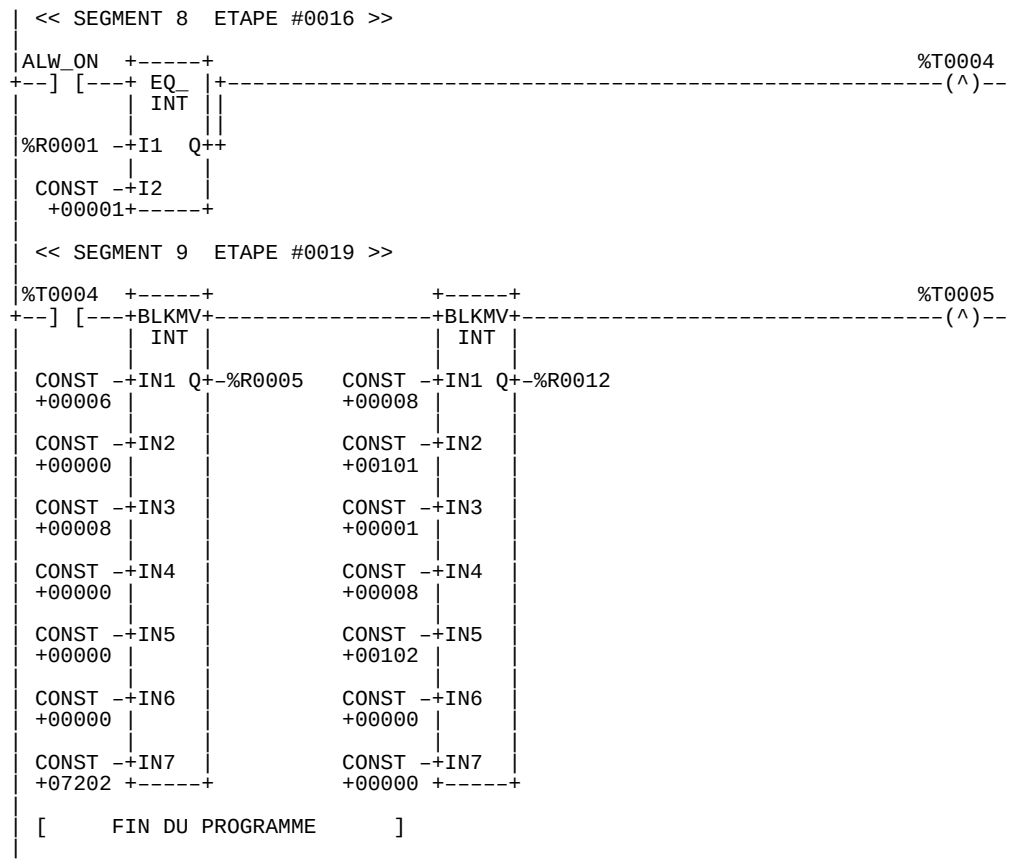
Dans le programme en diagramme en échelle fourni à titre d'exemple, les entrées de l'instruction COMREQ ont été définies comme suit :

- L'entrée IN désigne le Registre %R00005 comme début du bloc de commande.
- L'entrée SYSID indique que la commande est destinée au module CMM installé dans le rack 0, emplacement 2.
- L'entrée TASK indique que la commande est destinée au port 1 du module CMM.

Assurez–vous que les valeurs spécifiées dans SYSID et TASK correspondent à votre configuration système et que le port spécifié est bien configuré en mode SNP maître.



(suite à la page suivante)


| << SEGMENT 8 ETAPE #0016 >>
|ALW_ON +-----+   %T0004
+--] [---+ EQ_ |+---(^)--
| | INT ||
| | ||
|%R0001 -+I1 Q++
| | |
| CONST -+I2 |
| +00001+-----+
| << SEGMENT 9 ETAPE #0019 >>
|%T0004 +-----+   +-----+   %T0005
+--] [---+BLKMV+----------------+BLKMV+---------------------------------(^)--
| | INT | | INT |
| | | | |
| CONST -+IN1 Q+-%R0005   CONST -+IN1 Q+-%R0012
+00006		+00008	
CONST -+IN2	CONST -+IN2
+00000		+00101	
CONST -+IN3	CONST -+IN3
+00008		+00001	
CONST -+IN4	CONST -+IN4
+00000		+00008	
CONST -+IN5	CONST -+IN5
+00000		+00102	
CONST -+IN6	CONST -+IN6
+00000		+00000	
CONST -+IN7	CONST -+IN7
+07202 +-----+   +00000 +-----+
[FIN DU PROGRAMME   ]

3.2. EXEMPLES DE COMMANDES SNP

Le tableau ci–dessous liste toutes les commandes SNP destinées à un module CMM qui peuvent être générées par un COMREQ. La plupart d'entre elles ne peuvent être générées que sur le port série d'un module CMM configuré comme un équipement SNP maître. Les commandes déportées permettent à un équipement maître de communiquer avec des équipements esclaves. Sauf indication contraire, un équipement maître peut envoyer des commandes déportées à tout équipement Série 90 esclave ; certaines commandes déportées ne sont valides que lors d'une communication avec un équipement Série 90–70 esclave. Les commandes locales s'exécutent sans que l'équipement maître communique avec un équipement esclave.

Tableau 6-9. Commandes SNP

Commande SNP	Numéro de la commande		Maître	Esclave
	Décimal	Hexadécimal		
Commandes locales				
Effacer Mots d'état de diagnostic	7000	(1B58)	X	X
Lire Mots d'état de diagnostic	7001	(1B59)	X	X
Modifier ID SNP	7002	(1B5A)		X
Programmer Adresse des bits d'état X	7003	(1B5B)		X
Commandes SNP–X déportées				
Lire X	7101	(1BBD)	X	
Ecrire X	7102	(1BBE)	X	
Commandes SNP déportées				
Connecter	7200	(1C20)	X	
Modifier niveau de protection	7201	(1C21)	X	
Lire Mémoire système	7202	(1C22)	X	
Ecrire dans mémoire système	7203	(1C23)	X	
Lire Mémoire des tâches	7204	(1C24)	vers 90–70 seulement(1)	
Ecrire dans mémoire des tâches	7205	(1C25)	vers 90–70 seulement(1)	
Lire Mémoire des blocs du programme	7206	(1C26)	vers 90–70 seulement(1)	
Ecrire dans mémoire des blocs du programme	7207	(1C27)	vers 90–70 seulement(1)	
Etat court de l'API	7208	(1C28)	X	
Retourner Nom du programme de commande	7209	(1C29)	X	
Retourner Type et ID de l'API	7210	(1C2A)	X	
Retourner Heure/Date de l'API	7211	(1C2B)	X	
Retourner Table des défauts	7212	(1C2C)	X	
Programmer Heure/Date de l'API	7213	(1C2D)	X	
Forcer bascule de la mémoire système	7214	(1C2E)	X	
Etablir un datagramme	7215	(1C2F)	X	
Mettre à jour le datagramme	7216	(1C30)	X	
Annuler le datagramme	7217	(1C31)	X	
Mettre à jour le datagramme en temps réel	7218	(1C32)	X	
Connexion longue	7300	(1C84)	X	
Commandes spéciales				
Numérotation automatique	7400	(1CE8)	X	

1 Cette commande n'est valide que lors d'une communication avec un équipement Série 90–70 esclave.
Toutes les commandes SNP présentées dans ce paragraphe retournent le mot d'état COMREQ dans le registre %R00001.

3.2.1. Commande Effacer Mots d’état de diagnostic : 07000 (1B58)

Modes : Maître et Esclave

Description :

Commande locale. Cette commande efface le contenu des mots d’état de diagnostic SNP résidant dans le module CMM. Il existe un jeu complet de mots d’état de diagnostic pour chaque port du module CMM.

Exemple de bloc de commande :

Effacer les mots d’état de diagnostic du module CMM local correspondant au port série spécifié dans le champ TACHE de l’instruction COMREQ.

```
Mot 1 :    00001 (0001) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d’état (%R)
Mot 4 :    00000 (0000) Adresse du mot d’état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07000 (1B58) Numéro de la commande SNP
```

3.2.2. Commande Lire les mots d'état de diagnostic : 07001 (1B59)

Modes : Maître et Esclave

Description :

Commande locale. Cette commande retourne les mots d'état de diagnostic SNP résidant dans le module CMM dans la zone de mémoire de l'API spécifiée dans l'instruction COMREQ. Il existe un jeu complet de mots d'état de diagnostic pour chaque port du module CMM.

Exemple de bloc de commande :

Lire les 20 mots d'état de diagnostic locaux du module CMM et les placer dans les registres 181–200 de la mémoire des registres (%R). Commande destinée au port série spécifié dans le champ TACHE de l'instruction COMREQ.

```
Mot 1 :   00005 (0005) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07001 (1B59) Numéro de la commande SNP

Mot 8 :   00001 (0001) Premier mot d'état de diagnostic à lire (DSW 1)
Mot 9 :   00020 (0014) Nombre de mots d'état de diagnostic à lire (20 DSW)
Mot 10 :  00008 (0008) Type de mémoire dans lequel ils seront sauvegardés (%R)
Mot 11 :  00181 (00B5) Adresse dans laquelle ils seront sauvegardés (Registre 181)
```

Les champs "Type de mémoire et Adresse dans lequel/laquelle ils seront sauvegardés" doivent être définis en fonction du nombre de mots d'état de diagnostic à lire. Voir Tableau 6.1.

Mots d'état de diagnostic SNP :

Le tableau ci–dessous indique le contenu des différents mots d'état de diagnostic SNP :

Tableau 6-10. Définition des mots d'état de diagnostic SNP

Numéro du mot	Contenu
Mot 1	Mot d'état Erreur SNP
Mot 2	Nombre de demandes/réponses ayant abouti
Mot 3	Nombre de demandes erronées
Mot 4	Nombre d'arrêts prématurés
Mot 5	Nombre d'essais de retransmission
Mot 6	Nombre de NAK envoyés
Mot 7	Nombre de séquences Break reçues
Mots 8–13	Réservés (00h)
Mot 14	Version du logiciel SNP
Mots 15–20	6 premiers mots du bloc de données d'un COMREQ erroné

Mot d'état Erreur SNP. Ce mot renferme les codes d'erreurs majeures et mineures correspondant à la dernière erreur COMREQ ou erreur SNP interne. Son format et son contenu sont identiques à ceux du mot d'état SNP. (Pour plus d'informations à ce sujet, voir § 2, Mot d'état SNP).

Nombre de demandes/réponses ayant abouti. Ce mot indique le nombre d'instructions COMREQ ou de réponses SNP ayant abouti. Pour l'équipement maître, cette valeur indique le nombre de COMREQ qui se sont exécutés correctement ; elle inclut également tout message supplémentaire échangé pour maintenir la communication SNP active. Pour un équipement esclave, cette valeur indique le nombre de réponses SNP transmises avec succès à l'équipement maître déporté ; elle inclut également tout message envoyé pour maintenir la communication SNP active, ainsi que le nombre d'instructions COMREQ locales qui se sont exécutées correctement.

Nombre de demandes erronées. Ce mot indique le nombre d'instructions COMREQ qui n'ont pas abouti pour cet équipement SNP. Cette valeur n'indique que les erreurs non bloquantes ; la communication SNP n'est pas interrompue. Pour un équipement SNP esclave, cette valeur indique également le nombre de réponses erronées retournées à l'équipement maître.

Nombre d'arrêts prématurés. Ce mot indique le nombre d'erreurs bloquantes pour cet équipement SNP ; la communication SNP est interrompue. Un arrêt prématuré peut se produire tant sur l'équipement maître que sur un équipement esclave. Après un arrêt prématuré, l'équipement maître doit demander une nouvelle communication SNP. Les arrêts prématurés peuvent provenir de la non–réception d'un certain message ou d'un acquittement, de certaines erreurs de protocole, d'une erreur de lecture ou d'écriture sur le port série, ou d'une défaillance de la communication avec l'API local. Les temps limites d'inactivité, qui n'interrompent pas la communication SNP, ne sont pas comptés comme des arrêts prématurés ; un dépassement du temps limite d'inactivité peut survenir dans l'équipement esclave lorsque celui–ci ne reçoit plus d'autres messages de l'équipement maître.

Nombre d'essais de retransmission. Ce mot indique le nombre d'essais de retransmission d'un message par cet équipement SNP. Un équipement maître ou esclave essaie de retransmettre un message lorsque l'accusé de réception (ou acquittement) de ce message indique un erreur de transmission récupérable. Tout message SNP peut être retransmis deux fois au maximum ; si le message n'est toujours pas acquitté correctement, une erreur bloquante se produit et la communication SNP est interrompue. Sous SNP–X, ce mot indique le nombre d'essais de retransmission d'un message de connexion X pour ouvrir une session SNP–X, ainsi que toute demande SNP–X réitérée, après réouverture d'une session.

Nombre de NAK envoyés. Ce mot indique le nombre d'accusés de réception négatifs (NAK) retournés par cet équipement SNP. Un NAK est envoyé chaque fois qu'une erreur (récupérable ou bloquante) est détectée dans un message SNP reçu. Si l'erreur est récupérable, un nouvel essai est attendu ; si l'erreur est bloquante, la communication SNP est interrompue.

Nombre de séquences Break reçues. Ce mot indique le nombre de séquences Break reçues par cet équipement SNP. Il n'est utilisé que par les équipements esclaves. A réception d'une séquence Break, toute communication SNP en cours est interrompue. La séquence Break précède immédiatement chaque nouvelle demande de communication SNP.

Version du logiciel SNP. Ce mot indique la version du microprogramme de communication de ce module CMM. Le numéro de version est fourni sous forme de deux nombres hexadécimaux (octet de poids fort, octet de poids faible).

Bloc de données d'un COMREQ erroné. Ces six mots correspondent aux 6 premiers mots du bloc de données de la dernière instruction COMREQ qui n'a pas abouti. Ils sont utilisés à la fois par l'équipement maître et les équipements esclaves (les équipements esclaves n'exécutent que des commandes COMREQ locales). Ces données permettent d'identifier l'instruction COMREQ qui ne s'est pas exécutée correctement.

3.2.3. Commande Modifier ID SNP : 07002 (1B5A)

Mode : Esclave

Description :

Commande locale. L'équipement esclave dans le module CMM prend automatiquement le même ID SNP que celui de l'équipement esclave dans la CPU de l'API. Cette commande remplace l'ID SNP de l'équipement CMM esclave par la valeur spécifiée. Des ID SNP distincts doivent être définis pour différencier les différents équipements SNP esclaves dans une configuration multipoint.

L'équipement SNP esclave du module CMM reprend l'ID SNP par défaut (exemple, l'ID SNP de la CPU de l'API) chaque fois que l'API est remis sous tension ou que le module CMM est redémarré manuellement.

Exemple de bloc de commande :

Redéfinir l'ID SNP de l'équipement SNP esclave configuré dans le module CMM à "NEW ID1".

```
Mot 1 :   00005 (0005) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07002 (1B5A) Numéro de la commande SNP

Mot 8 :   17742 (454E) Caractères 1 et 2 de l'ID SNP esclave : N (4Eh), E (45h)
Mot 9 :   18775 (4957) Caractères 3 et 4 de l'ID SNP esclave : W (57h), I (49h)
Mot 10 :  12612 (3144) Caractères 5 et 6 de l'ID SNP esclave : D (44h), 1 (31h)
Mot 11 :  00000 (0000) Caractères 7 et 8 de l'ID SNP esclave : nul, nul
```

Le champ "ID SNP esclave" spécifie le nouvel ID SNP de l'équipement esclave. Ce champ comporte 8 octets. Pour les API Série 90–70, l'ID SNP doit comporter 7 octets maximum suivis d'un caractère nul (0) et accepte tous les caractères ASCII. Pour les API Série 90–30, l'ID SNP est limité à 6 octets maximum suivis d'un caractère nul (0) ; les 6 octets ne peuvent renfermer que les caractères ASCII '0' à '9' compris et les majuscules 'A' à 'F' comprises. L'ID SNP nul (Caractère 1 = 0) n'est pas accepté.

Remarque

La première instruction COMREQ générée sur un module CMM après la mise sous tension de l'API doit être retardée ; le module CMM n'accepte pas les COMREQ au démarrage du programme d'application API. Une temporisation de 2 secondes après la première scrutation est recommandée.

3.2.4. Commande Programmer adresse des bits d'état X : 07003 (1B5B)

Mode : Esclave

Description :

Commande locale. Cette commande spécifie la mémoire binaire de l'API local qui sera mis à jour par l'équipement esclave lors d'une opération SNP–X. Trois bits de mémoire contigus, appelés bits d'état X, indiquent l'utilisation d'une commande X par l'équipement esclave. Les bits d'état X peuvent être examinés à tout instant à partir du programme en diagramme en échelle de l'API. Les bits d'état X de l'équipement CMM esclave ne sont mis à jour que lorsqu'une adresse mémoire valide de l'API a été spécifiée via cette commande. (Si une erreur se produit pendant l'exécution de ce COMREQ, l'adresse des bits d'état X reste indéfinie ; les bits d'état X ne sont mis à jour dans la mémoire de l'API local que lorsque l'instruction COMREQ a abouti).

Les bits d'état X utilisés par l'équipement esclave dans le module CMM sont similaires à ceux définis dans la mémoire %S (%S17–%S19) et qui sont utilisés par l'équipement esclave sur le port série interne des API Série 90–20 et Série 90–30. (Voir Note ci–après).

Les bits d'état X consistent en trois bits contigus. L'adresse du bit de poids le plus faible (X_ACTIVE) est spécifiée dans le bloc de données COMREQ ; les deux autres bits d'état X occupent toujours les deux adresses mémoire qui suivent. Les bits d'état X ne peuvent être définis que dans les types de mémoire orientés Bit %I, %Q, %T, ou %M.

L'exemple qui suit définit les bits d'état X dans les bits 101–103 de la mémoire interne logique (%M, mode bit).

```
Mot 1 :   00003 (0003) Longueur du bloc de données SNNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07003 (1B5B) Numéro de la commande SNP–X

Mot 8 :   00076 (004C) Type de mémoire esclave des bits d'état X (%M, mode bit)
Mot 9 :   00101 (0065) Adresse esclave des bits d'état X (%M101)
```

Les champs "Type de mémoire et Adresse esclave des bits d'état X" doivent être définis de façon à pouvoir contenir les trois bits d'état X. Vous ne pouvez sélectionner que les types de mémoire orientés Bit %I, %Q, %M, ou %T ; voir Table des types de mémoire (Table 15). Si l'adresse des bits d'état X n'est pas spécifiée (le type de mémoire et l'adresse sont tous deux définis à 0), les données contenues dans les bits d'état X ne sont pas écrites dans l'API.

Les bits d'état X sont configurés comme suit :

(bit n+2)	(bit n+1)	(bit n)
X_WRITE	X_READ	X_ACTIVE

Le bit X_ACTIVE indique qu'une session de communication SNP–X est active ; en d'autres termes, cet équipement esclave peut recevoir des demandes X de transmission de données pendant que ce bit est à 1. Ce bit est à 1 chaque fois qu'une nouvelle session de communication SNP–X est établie ; il est mis à 0 chaque fois qu'un Break est reçu ou qu'une erreur bloquante se produit.

Les bits X_READ et X_WRITE indiquent qu'une demande X déportée a été satisfaite par cet équipement esclave. Le bit approprié est mis à 1 après chaque demande X de lecture ou d'écriture qui a abouti. Le programme en diagramme en échelle de l'API doit détecter les bits X_READ et X_WRITE à chaque cycle. Une fois détecté, chaque bit doit immédiatement être mis à 0 pour pouvoir détecter correctement la prochaine demande X déportée ayant abouti.

Remarque

Les bits d'état X opèrent différemment pour les équipements esclaves sur les modules CMM et ceux sur le port série interne de la CPU de l'API. Pour la CPU de l'API, les bits d'état X sont dans des adresses mémoires prédéfinies et sont toujours mis à jour après chaque opération de l'équipement SNP–X esclave. Les bits X_READ et X_WRITE restent à 1 pendant un cycle complet et ne peuvent pas être mis à 0 par l'application en diagramme en échelle de l'API.

3.2.5. Commande Lire X : 07101 (1BBD)

Mode : Maître

Description :

Commande déportée. L'équipement maître établit une nouvelle session de communication SNP–X avec l'équipement esclave si aucune session appropriée n'est déjà active. L'équipement maître envoie alors une demande de Lecture X à l'équipement esclave qui répond en envoyant des données. Ce service est fourni pour permettre un accès rapide en lecture aux diverses tables de références de l'API esclave. La commande lit le volume de données correspondant au nombre d'éléments spécifié dans la table de références de l'équipement maître. Lorsque les types de mémoire des tables de références des équipements maître et esclave(s) diffèrent, les données sont transférées depuis la table de références de l'équipement esclave en commençant par le bit le moins significatif. La commande Lire X ne doit pas utiliser l'ID SNP de diffusion.

Cette commande X nécessite au moins 11 mots dans le bloc de données COMREQ. De 2 à 6 mots supplémentaires peuvent être spécifiés pour fournir des paramètres optionnels.

L'exemple qui suit établit une mono–session de communication directe vers l'équipement esclave via l'ID SNP nul, puis lit le registre 201 de la mémoire des registres (%R) de l'équipement esclave et copie les 10 bits les moins significatifs dans les entrées 1–10 de la mémoire des entrées (%I) de l'équipement maître. Le mot d'état de l'API retourné par l'équipement esclave est mis à jour dans le registre 32 de la mémoire des registres (%R) de l'équipement maître.

```
Mot 1 :   00017 (0011) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse mémoire du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07101 (1BBD) Numéro de la commande SNP–X

Mot 8 :   00000 (0000) Caractères 1 et 2 de l'ID SNP esclave : nul, nul
Mot 9 :   00000 (0000) Caractères 3 et 4 de l'ID SNP esclave : nul, nul
Mot 10 :  00000 (0000) Caractères 5 et 6 de l'ID SNP esclave : nul, nul
Mot 11 :  00000 (0000) Caractères 7 et 8 de l'ID SNP esclave : nul, nul
Mot 12 :  00000 (0000) Type de session de communication (0 = mono–session ;
                       1 = multi–session)
Mot 13 :  00008 (0008) Type de mémoire esclave à partir duquel les données seront lues(%R)
Mot 14 :  00201 (00C9) Adresse esclave à partir de laquelle les données seront lues(Registre
                       201)

Mot 15 :  00010 (000A) Nombre d'éléments à lire dans type de mémoire maître
Mot 16 :  00070 (0046) Type de mémoire maître dans lequel les données seront sauvegardées
                       (%I)
Mot 17 :  00001 (0001) Adresse maître dans laquelle les données seront sauvegardées
                       (Entrée 1)
Mot 18 :  00008 (0008) Type de mémoire maître dans lequel le mot d'état esclave sera
                       sauvegardé (%R)
Mot 19 :  00032 (0020) Adresse maître dans laquelle le mot d'état de l'API esclave sera
                       sauvegardé (Registre 32)
Mot 20 :  00000 (0000) Temps limite de réponse (en ms ; 0 = valeur par défaut)
Mot 21 :  00000 (0000) Délai de diffusion (en ms ; 0 = valeur par défaut)

Mot 22 :  00000 (0000) Temps de retournement (Turnaround) du modem (en ms)
Mot 23 :  00000 (0000) Délai de transmission (en ms)
```

Le champ ”ID SNP” spécifie l’équipement esclave duquel les données seront extraites. L’ID SNP nul (tous les octets à 0h) ne peut être utilisé que lorsque le type de session de communication Mono est sélectionné. L’ID SNP de diffusion (tous les octets à **FFh**) ne peut pas être utilisé par cette commande.

Le champ ”Type de session de communication” spécifie s’il s’agit d’une mono–session ou d’une multi–session de communication. L’équipement maître utilise cette valeur, ainsi que la valeur ID SNP, pour déterminer si la session de communication appropriée est déjà active, ou si une nouvelle session doit être établie avant le transfert des données. En mode mono–session, la communication n’est établie qu’avec l’équipement esclave spécifié par l’ID SNP. En mode multi–session, la communication est établie avec tous les équipements esclaves sur la liaison série. Dans les deux cas, une fois la communication établie, l’équipement maître envoie la commande de transfert des données à l’équipement esclave spécifié par l’ID SNP. Lorsque l’équipement maître ne communique qu’avec un seul équipement esclave, le mode Mono–session fournit la performance la plus rapide. Lorsqu’il communique avec tous les équipements esclaves sur une liaison série multipoint, le mode Multi–session fournit la performance la plus rapide.

Le champ ”Nombre d’éléments à lire dans type de mémoire maître” doit être défini en fonction des unités correspondant au type de mémoire de l’équipement maître. 1000 octets de données maximum peuvent être transférés avec une seule commande Lire X ; si le volume des données à transférer est plus important, vous pouvez utiliser plusieurs commandes.

Remarque

Les paramètres suivants sont optionnels. La Longueur du bloc de données COMREQ spécifiée (Mot 1) doit inclure tous les paramètres utilisés.

Les champs optionnels ”Type et Adresse mémoire maître dans lequel/laquelle le mot d’état esclave sera sauvegardé” indiquent l’adresse mémoire dans l’API maître dans laquelle le mot d’état de l’API sera mis à jour depuis l’équipement esclave, une fois que la commande aura abouti. Si ces deux champs ne sont pas programmés, ou s’ils sont définis à zéro, le mot d’état de l’API esclave ne sera pas mis à jour dans la mémoire de l’API maître.

Le champ optionnel ”Temps limite de réponse” modifie la valeur du temporisateur Temps Limit. Il détermine le laps de temps maximum pendant lequel l’équipement maître attendra une réponse X de la part de l’équipement esclave. La nouvelle valeur doit être spécifiée en millisecondes et n’est utilisée que pour cette instruction COMREQ. Si ce champ n’est pas programmé, ou s’il est défini à zéro, l’équipement maître utilise la valeur du temporisateur T2, définie par le paramètre de configuration **Temps Limit**.

Le champ optionnel ”Délai de diffusion” modifie la valeur du temporisateur Délai de diffusion. Il détermine le laps de temps maximum pendant lequel l’équipement maître attendra que tous les équipements esclaves établissent une session de communication SNP–X, ou pour traiter une commande de diffusion X, étant donné que les équipements esclaves ne répondent pas à un message de diffusion. La nouvelle valeur doit être spécifiée en millisecondes et n’est utilisée que pour cette instruction COMREQ. Si ce champ n’est pas programmé, ou s’il est défini à zéro, l’équipement maître utilise la valeur du temporisateur T2, définie par le paramètre de configuration **Temps Limit**.

Le champ optionnel ”Temps de retournement du modem” (Turnaround) modifie le temps de retournement du modem. Il détermine le laps de temps requis par un modem connecté pour retourner la liaison. La nouvelle valeur doit être spécifiée en millisecondes et n’est utilisée que pour cette instruction COMREQ. Si ce champ n’est pas programmé, ou s’il est défini à zéro, l’équipement maître utilise la valeur définie par le paramètre de configuration **Délai TurnA**.

Le champ optionnel ”Délai de transmission” modifie le délai de transmission. Il tient compte des délais de transmission inhabituellement longs entre l’équipement maître et l’équipement esclave. Les délais inhabituellement longs sont typiques des liaisons par satellite. La nouvelle valeur doit être spécifiée en millisecondes et n’est utilisée que pour cette instruction COMREQ. Si ce champ n’est pas programmé, ou s’il est défini à zéro, l’équipement maître utilise la valeur définie par le paramètre de configuration **Temps Limit**.

3.2.6. Commande Ecrire X : 07102 (1BBE)

Mode : Maître

Description :

Commande déportée. L'équipement maître établit une nouvelle session de communication SNP–X avec l'équipement esclave si aucune session appropriée n'est déjà active. L'équipement maître envoie alors une demande d'Ecriture X accompagnée de données à l'équipement esclave. Ce service est fourni pour permettre un accès rapide en écriture dans les diverses tables de références de l'API esclave. La commande écrit le nombre d'éléments spécifié, de la table de références de l'équipement maître dans celle de l'équipement esclave. Lorsque les types de mémoire des tables de références des équipements maître et esclave(s) diffèrent, les données sont complétées par autant de 0 que nécessaire. En mode Multi–session, l'ID SNP de diffusion peut être utilisé pour écrire des données dans tous les équipements esclaves sur la liaison série.

Cette commande X nécessite au moins 11 mots dans le bloc de données COMREQ. De 2 à 6 mots supplémentaires peuvent être spécifiés pour fournir des paramètres optionnels.

L'exemple qui suit établit une mono–session de communication directe vers l'équipement esclave via l'ID SNP nul, puis écrit les entrées 1–10 de la mémoire des entrées (%I) de l'équipement maître dans le registre 201 de la mémoire des registres (%R) de l'équipement esclave. Le mot d'état de l'API, retourné par l'équipement esclave, est mis à jour dans le registre 32 de la mémoire des registres (%R) de l'équipement maître.

```
Mot 1 :   00017 (0011) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07102 (1BBE) Numéro de la commande SNP–X

Mot 8 :   00000 (0000) Caractères 1 et 2 de l'ID SNP esclave : nul, nul
Mot 9 :   00000 (0000) Caractères 3 et 4 de l'ID SNP esclave : nul, nul
Mot 10 :  00000 (0000) Caractères 5 et 6 de l'ID SNP esclave : nul, nul
Mot 11 :  00000 (0000) Caractères 7 et 8 de l'ID SNP esclave : nul, nul
Mot 12 :  00000 (0000) Type de session de communication (0 = mono–session ;
                       1 = multi–session)
Mot 13 :  00008 (0008) Type de mémoire esclave dans lequel les données seront sauvegardées
                       (%R)
Mot 14 :  00201 (00C9) Adresse esclave dans laquelle les données seront sauvegardées
                       (Registre 201)

Mot 15 :  00010 (000A) Nombre d'éléments à écrire dans type de mémoire maître
Mot 16 :  00070 (0046) Type de mémoire maître à partir duquel les données seront écrites
                       (%I)
Mot 17 :  00001 (0001) Adresse maître à partir de laquelle les données seront écrites
                       (Entrée 1)
Mot 18 :  00008 (0008) Type de mémoire maître dans lequel le mot d'état de l'API esclave
                       sera sauvegardé (%R)
Mot 19 :  00032 (0020) Adresse maître dans laquelle le mot d'état de l'API esclave sera
                       sauvegardé (Registre 32)
Mot 20 :  00000 (0000) Temps limite de réponse (en ms ; 0 = valeur par défaut)
Mot 21 :  00000 (0000) Délai de diffusion (en ms ; 0 = valeur par défaut)

Mot 22 :  00000 (0000) Temps de retournement (Turnaround) du modem (en ms)
Mot 23 :  00000 (0000) Délai de transmission (en ms)
```

Le champ "ID SNP" spécifie le ou les équipements esclaves dans lesquels les données seront écrites. L'ID SNP nul (tous les octets à 0h) ne peut être utilisé que lorsque le type de session de communication Mono est sélectionné. L'ID SNP de diffusion (tous les octets à **FFh**) ne peut être utilisé que lorsque le type de session de communication Multi est sélectionné.

Le champ "Type de session de communication" spécifie s'il s'agit d'une mono–session ou d'une multi–session de communication. L'équipement maître utilise cette valeur, ainsi que la valeur ID SNP, pour déterminer si la session de communication appropriée est déjà active, ou si une nouvelle session doit être établie avant le transfert des données. En mode mono–session, la communication n'est établie qu'avec l'équipement esclave spécifié par l'ID SNP. En mode multi–session, la communication est établie avec tous les équipements esclaves sur la liaison série. Dans les deux cas, une fois la communication établie, l'équipement maître envoie la commande de transfert des données à l'équipement esclave spécifié par l'ID SNP. Lorsque l'équipement maître ne communique qu'avec un seul équipement esclave, le mode Mono–session fournit la performance la plus rapide. Lorsqu'il communique avec tous les équipements esclaves sur une liaison série multipoint, le mode Multi–session fournit la performance la plus rapide.

Le champ "Nombre d'éléments à écrire dans type de mémoire maître" doit être défini en fonction des unités correspondant au type de mémoire de l'équipement maître. 1000 octets de données maximum peuvent être transférés avec une seule commande Ecrire X ; si le volume des données à transférer est plus important, vous pouvez utiliser plusieurs commandes.

Remarque

Lorsque les types de données ne concordent pas, les valeurs sont remplacées par des zéros.

Les paramètres COMREQ suivants sont optionnels. La Longueur du bloc de données COMREQ (Mot 1) doit inclure tous les paramètres utilisés.

Les champs optionnels "Type et Adresse mémoire maître dans lequel/laquelle le mot d'état de l'API esclave sera sauvegardé" indiquent l'adresse mémoire dans l'API maître dans laquelle le mot d'état de l'API sera mis à jour depuis l'équipement esclave, une fois que la commande aura abouti. Si ces deux champs ne sont pas programmés, ou s'ils sont définis à zéro, le mot d'état de l'API esclave ne sera pas mis à jour dans la mémoire de l'API maître.

Le champ optionnel "Temps limite de réponse" modifie la valeur du temporisateur Temps Limit. Il détermine le laps de temps maximum pendant lequel l'équipement maître attendra une réponse X de la part de l'équipement esclave. La nouvelle valeur doit être spécifiée en millisecondes et n'est utilisée que pour cette instruction COMREQ. Si ce champ n'est pas programmé, ou s'il est défini à zéro, l'équipement maître utilise la valeur du temporisateur T2, définie par le paramètre de configuration **Temps Limit**.

Le champ optionnel "Délai de diffusion" modifie la valeur du temporisateur Délai de diffusion. Il détermine le laps de temps maximum pendant lequel l'équipement maître attendra que tous les équipements esclaves établissent une session de communication SNP–X, ou pour traiter une commande de diffusion X, étant donné que les équipements esclaves ne répondent pas à un message de diffusion. La nouvelle valeur doit être spécifiée en millisecondes et n'est utilisée que pour cette instruction COMREQ. Si ce champ n'est pas programmé, ou s'il est défini à zéro, l'équipement maître utilise la valeur du temporisateur T2, définie par le paramètre de configuration **Temps Limit**.

Le champ optionnel "Temps de retournement du modem" (Turnaround) modifie le temps de retournement du modem. Il détermine le laps de temps requis par un modem connecté pour retourner la liaison. La nouvelle valeur doit être spécifiée en millisecondes et n'est utilisée que pour cette instruction COMREQ. Si ce champ n'est pas programmé, ou s'il est défini à zéro, l'équipement maître utilise la valeur définie par le paramètre de configuration **Délai TurnA**.

Le champ optionnel "Délai de transmission" modifie le délai de transmission. Il tient compte des délais de transmission inhabituellement longs entre l'équipement maître et l'équipement esclave. Les délais inhabituellement longs sont typiques des liaisons par satellite. La nouvelle valeur doit être spécifiée en millisecondes et n'est utilisée que pour cette instruction COMREQ. Si ce champ n'est pas programmé, ou s'il est défini à zéro, l'équipement maître utilise la valeur définie par le paramètre de configuration **Temps Limit**.

3.2.7. Commande de connexion : 07200 (1C20)

Mode : Maître

Description :

Commande déportée. La commande de connexion établit une session de communication avec un équipement esclave. Elle peut être générée à tout instant. L'équipement maître envoie une séquence Break, suivie d'une demande de connexion ; l'équipement esclave spécifié renvoie une réponse de connexion. La séquence Break est adressée à tous les équipements esclaves partageant la même liaison série. La détection d'une séquence Break par un équipement esclave arrête immédiatement toute session de communication en cours et place tous les équipements esclaves en état d'attente d'une demande de connexion valide. L'équipement maître attend que le laps de temps défini par le temporisateur T4 ait expiré pour envoyer la demande de connexion aux équipements esclaves. Seul l'équipement esclave dont l'ID SNP correspond retournera une réponse de connexion. Les autres reviennent automatiquement à l'état d'attente d'une séquence Break. A réception d'une réponse de connexion valide, l'équipement maître complète l'instruction COMREQ avec le code Connexion réussie. S'il ne reçoit pas de réponse, ou que la réponse est invalide, l'équipement maître complète l'instruction COMREQ avec un code d'erreur.

Une fois connecté à un API esclave, l'équipement maître bénéficie automatiquement du niveau de protection défini par défaut dans l'équipement esclave. Pour le port interne de la CPU d'un API Série 90–70, le niveau de protection par défaut des équipements esclaves est le Niveau 0 ; le niveau 0 protège la mémoire de l'API en lecture et en écriture. Pour le port interne de la CPU d'un API Série 90–30 ou Série 90–20, le niveau de protection par défaut des équipements esclaves est le Niveau 1 ; le niveau 1 autorise la lecture de la mémoire de l'API mais interdit tout accès en écriture. Pour un équipement SNP esclave sur le module CMM d'un API Série 90, le niveau de protection est toujours le Niveau 2 et il ne peut pas être modifié ; le niveau 2 autorise les accès en lecture et en écriture dans la mémoire de l'API.

Si l'équipement maître requiert d'autres niveaux de protection, voir la commande Modifier le niveau de protection, plus loin dans ce chapitre.

La commande de connexion valide ou invalide également la signalisation de l'état complémentaire pendant la durée de la communication SNP. Lorsqu'il est validé, l'état complémentaire est mis à jour après chaque commande ayant abouti.

Si une installation particulière exige que les paramètres soient définis à des valeurs autres que celles fournies par le logiciel Logicmaster 90 ou la miniconsole de programmation HHP, vous devez utiliser la commande Connexion longue. Cette commande rallonge le temps système du processus de communication et ne doit par conséquent être utilisée que lorsque nécessaire. Au nombre des cas susceptibles de nécessiter une commande de connexion longue, citons :

- Liaisons par satellites
- Liaisons par modem avec des temps d'établissement > 500ms
- Communications en environnements perturbés
- Besoins spécifiques en matière de détection d'erreur/reprise après incident

Exemple de bloc de commande :

Se connecter à un équipement esclave dont l'ID SNP est "SNP ID1". Valider la mise à jour de l'état complémentaire sur cette commande et les suivantes ; enregistrer les 6 octets de l'état complémentaire dans les registres 171–173 de la mémoire des registres (%R) de l'équipement maître.

```
Mot 1 :   00007 (0007) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire pour mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07200 (1C20) Numéro de la commande SNP

Mot 8 :   20051 (4E53) Caractères 1 et 2 de l'ID SNP esclave : S (53h), N (4Eh)
Mot 9 :   18768 (4950) Caractères 3 et 4 de l'ID SNP esclave : P (50h), I (49h)
Mot 10 :  12612 (3144) Caractères 5 et 6 de l'ID SNP esclave : D (44h), 1 (31h)
Mot 11 :  00000 (0000) Caractères 7 et 8 de l'ID SNP esclave : nul, nul
Mot 12 :  00008 (0008) Type de mémoire maître dans lequel l'état complémentaire sera
                       sauvegardé (%R)
Mot 13 :  00171 (00AB) Adresse maître dans laquelle l'état complémentaire sera
                       sauvegardé (Registre 171)
```

Le champ "ID SNP esclave" spécifie l'ID SNP de l'équipement esclave désiré. Ce champ comporte 8 octets. Pour les API Série 90–70, l'ID SNP peut comporter 7 octets maximum suivis d'un caractère nul (0) et accepte tous les caractères ASCII. Pour les API Série 90–30, l'ID SNP est limité à 6 octets maximum suivis d'un caractère nul (0) ; les 6 octets ne peuvent renfermer que les caractères ASCII '0' à '9' compris et les majuscules 'A' à 'F' comprises.

L'ID SNP nul (caractère 1 = 0) peut être utilisé par l'équipement maître pour se connecter à n'importe quel équipement SNP esclave, indépendamment de l'ID SNP qui lui a été affecté. Les commandes avec des ID SNP nuls ne peuvent aboutir *que* dans une configuration point–à–point. Dans une configuration multipoint, tous les équipements SNP esclaves répondront à une commande de connexion spécifiant un ID SNP nul. Cette procédure donne toutefois des résultats imprévisibles et ne doit pas être tentée.

Les champs Type et Adresse mémoire maître dans lequel/laquelle l'état complémentaire sera sauvegardé sont optionnels. Si le Type et l'Adresse mémoire maître dans lequel/laquelle l'état complémentaire sera sauvegardé sont définis à zéro (aucune zone complémentaire n'est spécifiée), les données relatives à l'état complémentaire sont ignorées et ne sont pas fournies à l'application en diagramme en échelle de l'API. Si une adresse est spécifiée, la zone qui renfermera les 6 octets de l'état complémentaire décrit ci–dessous doit être définie. Les données relatives à l'état complémentaire sont retournées avec chaque message de réponse SNP transmis par l'équipement esclave. Une fois la zone d'état complémentaire spécifiée, c'est là que seront enregistrées toutes les données relatives à l'état complémentaire chaque fois qu'un message de réponse SNP Connexion réussie est reçu et ce, tant que la connexion reste établie. Voir dans le Tableau 6.1 les types et adresses mémoires valides.

Etat complémentaire :

Le tableau ci–après décrit les données relatives à l'état complémentaire :

Tableau 6-11. Données d'état complémentaires

Zone de mémorisation de l'état complémentaire		Description
Mot	**Octet**	
Mot 1 (octet de poids faible)	Octet 1	Numéro du programme de commande
Mot 1 (octet de poids fort)	Octet 2	Niveau de protection courant
Mot 2 (octet de poids faible)	Octet 3	Dernier temps de cycle
Mot 2 (octet de poids fort)	Octet 4	Dernier temps de cycle
Mot 3 (octet de poids faible)	Octet 5	Mot d'état API esclave
Mot 3 (octet de poids fort)	Octet 6	Mot d'état API esclave

Numéro de programme de commande. Cette valeur indique le numéro de la tâche dans le programme de commande auquel l'équipement SNP maître est couramment connecté. Compte tenu de l'actuelle mise en oeuvre, les valeurs valides sont –1 et 0 comme suit :

–1	L'équipement SNP maître n'est connecté à aucune tâche de programme de commande.
0	L'équipement SNP maître est connecté à la tâche 0 du programme de commande.

Niveau de protection. Niveau de protection courant de l'équipement SNP maître. Vous avez le choix entre 0 et 4 pour les API Série 90–70, et entre 1 et 4 pour les API Série 90–30.

Dernier temps de cycle. Cette valeur indique la durée du dernier cycle complet correspondant à la tâche du programme de commande principal. Elle est incrémentée par tranche de 100 microsecondes et est mesurée de Début de cycle (X–1) à Début de cycle (X).

Mot d'état API esclave. Les bits de ce mot sont définis dans le tableau qui suit, où le bit 0 est le bit de poids faible et le bit 15, le bit de poids fort.

Conformément à la convention d'écriture de l'ordre des octets dans un mot, l'octet 5 est l'octet de poids faible (renfermant les bits 0 à 7) et l'octet 6, l'octet de poids fort (renfermant les bits 8 à 15), comme illustré ci–dessous :

Octet 6	Octet 5
Bits 15–8	Bits 7–0

La signification de chaque bit du mot d'état de l'API est décrite dans le tableau qui suit :

Tableau 6-12. Données du mot d'état de l'API esclave

Numéro du bit	Description
Bit 0	Indicateur de dépassement de temps de cycle ; n'est significatif que lorsque le mode Cycle constant est actif. 1 = Dépassement de la valeur spécifiée pour Cycle constant. 0 = Aucune condition de dépassement de temps de cycle n'a été spécifiée.
Bit 1	Mode Cycle constant. 1 = Mode Cycle constant actif. 0 = Mode Cycle constant non actif.
Bit 2	Table des défauts de automate depuis dernière lecture par cet équipement. 1 = Table des défauts automate changée depuis dernière lecture. 0 = Table des défauts automate inchangée depuis dernière lecture.
Bit 3	Table des défauts d'E/S depuis dernière lecture par cet équipement. 1 = Table des défauts d'E/S changée depuis dernière lecture. 0 = Table des défauts d'E/S inchangée depuis dernière lecture.
Bit 4	Entrées présentes dans la table des défauts automate. 1 = Une ou plusieurs entrées présentes dans la table des défauts automate. 0 = Table des défauts automate vide.
Bit 5	Entrées présentes dans la table des défauts d'E/S. 1 = Une ou plusieurs entrées présentes dans la table des défauts d'E/S. 0 = Table des défauts d'E/S vide.
Bit 6	Indicateur de connexion d'une console de programmation. 1 = Connexion d'une console de programmation trouvée. 0 = Aucune connexion d'une console de programmation trouvée.
Bit 7	Configuration du commutateur ENABLE/DISABLE de la façade avant. 1 = Sorties activées. 0 = Sorties désactivées.
Bit 8	Configuration du commutateur RUN/STOP de la façade avant. 1 = RUN, 0 = STOP
Bit 9	Bit de protection OEM. 1 = Protection OEM en vigueur. 0 = Pas de protection OEM.
Bit 10	Non utilisé.
Bit 11	Non utilisé.
Bits 12–15	Etat de l'API : 0 = Run E/S activé. 1 = Run E/S désactivé. 2 = Stop E/S désactivé. 3 = CPU en défaut. 4 = CPU arrêtée. 5 = CPU suspendue. 6 = Stop E/S activé.

3.2.8. Commande Modifier le niveau de protection : 07201 (1C21)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande de modification du niveau de protection à laquelle l'équipement esclave répond. Ce service permet à l'équipement maître de modifier son niveau de protection d'accès à l'API esclave, sous réserve que le mot de passe soit fourni. Cette commande n'est utilisée que lorsque l'équipement maître ne dispose pas d'un niveau de protection suffisant à la suite d'une commande de connexion.

Exemple de bloc de commande :

Faire passer le niveau de protection à 2 sur l'équipement esclave connecté. Le mot de passe de l'équipement esclave est "PASS1".

```
Mot 1 :    00006 (0006) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07201 (1C21) Numéro de la commande SNP

Mot 8 :    00002 (0002) Niveau de protection demandé
Mot 9 :    16720 (4150) Mot de passe (octets 1 et 2) : P (50h), A (41h)
Mot 10 :   21331 (5353) Mot de passe (octets 3 et 4) : S (53h), S (53h)
Mot 11 :   00049 (0031) Mot de passe (octets 5 et 6) : 1 (31h)
Mot 12 :   00000 (0000) Mot de passe (octets 7 et 8) : nul, nul
```

Le champ "Niveau de protection demandé" renferme une valeur comprise entre 0 et 4 pour les API Série 90–70, et entre 1 et 4 pour les API Série 90–30 ou Série 90–20. Le niveau de protection peut également être défini à –1. Lorsque –1 est spécifié, le niveau de protection le plus élevé est accordé au demandeur, sous réserve que le mot de passe correspondant soit fourni.

Le champ "Mot de passe" spécifie le mot de passe requis pour accéder au niveau de protection désiré. Si vous n'entrez pas de mot de passe, mettez tous les octets de ce champ à zéro (0).

Remarque

Pour pouvoir fonctionner correctement, le module CMM doit bénéficier du niveau de protection 2 sur la CPU de l'API. Pour éviter tout risque d'incompatibilité avec des mises en oeuvres existantes dans l'équipement SNP maître, un équipement SNP esclave dans le module CMM acceptera tous les niveaux de protection et mots de passe figurant dans une demande de Modification du niveau de protection ; l'équipement CMM esclave restera toujours au niveau de protection 2.

Le tableau ci–après décrit la signification de chaque niveau de protection :

Tableau 6-13. Description des niveaux de protection de l'API

Niveau	Signification
Niveau 4	Autorisation d'écrire dans toutes les configurations ou tous les programmes. Les configurations ne peuvent être modifiées qu'en mode STOP ; les programmes peuvent être modifiés en mode STOP ou RUN. L'affichage, la création ou la suppression de mots de passe sont autorisés à tous les niveaux.
Niveau 3	Autorisation d'écrire dans toutes les configurations ou tous les programmes. Les modifications de même taille sont autorisées, de même que la création/suppression de programmes et le remplacement des E/S logiques.
Niveau 2	Autorisation d'écrire dans toute mémoire de données, sauf s'il s'agit de remplacer les E/S logiques. L'API peut être démarré ou arrêté. Les tables des défauts automate et d'E/S peuvent être effacées.
Niveau 1	Autorisation de lire toutes les mémoires de données. Les accès en écriture sont interdits. L'API ne peut pas être démarré ou arrêté.
Niveau 0	(API Série 90–70 seulement). Interdiction de lire ou d'écrire dans la mémoire du système d'API.

Le tableau ci–après indique le niveau de protection minimum requis dans l'équipement esclave pour que l'équipement SNP maître puisse exécuter chaque demande :

Tableau 6-14. Niveaux de protection requis dans l'API

Demande de service	Niveau de protection minimum	
	API Série 90–30 ou Série 90–20	API Série 90–70
Modifier Niveau de protection	1	0
Lire Mémoire système	1	1
Lire Mémoire des tâches	n/a	1
Lire Mémoire des blocs du programme	n/a	1
Ecrire dans Mémoire système	2	2
Ecrire dans Mémoire des tâches	n/a	2
Ecrire dans Mémoire des blocs du programme	n/a	2
Forcer Bascule de la mémoire système	2	2
Retourner Table des défauts	1	1
Retourner Nom du programme de commande	1	0
Retourner Type et ID de l'API	1	1
Retourner Heure/Date de l'API	1	1
Etat court de l'API	1	0
Programmer Heure/Date de l'API	2(1)	2
Etablir un datagramme	1	1
Mettre à jour le datagramme	1	1
Annuler le datagramme	1	1
Mettre à jour le datagramme en temps réel	1	0

[1] Pour les API Série 90–30 dotés d'un microprogramme CPU version 1.x, le niveau de protection 3 est requis.

3.2.9. Commande Lire Mémoire système : 07202 (1C22)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande de lecture à laquelle l'équipement esclave répond par des données. Ce service est fourni pour permettre l'accès aux diverses tables de références résidant dans l'API esclave. (Voir dans le Tableau 6.1 les différents types de mémoire supportés). La commande lit suffisamment de données dans la table de références de l'équipement esclave pour remplir le nombre d'éléments spécifié dans la table de références de l'équipement maître. Lorsque les types de mémoire des tables de références des équipements maître et esclave(s) diffèrent, les données sont transférées depuis la table de références de l'équipement esclave en commençant par le bit de poids faible.

Exemple de bloc de commande :

Lire le registre 1 de la mémoire des registres (%R) de l'équipement esclave connecté et placer les 10 bits de poids faibles dans les entrées 1–10 de la mémoire des entrées (%I) de l'équipement maître.

```
Mot 1 :   00006 (0006) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 7 :   07202 (1C22) Numéro de la commande SNP

Mot 8 :   00008 (0008) Type de mémoire esclave à partir duquel les données seront lues (%R)
Mot 9 :   00001 (0001) Adresse mémoire esclave à partir de laquelle les données seront
                       lues (Registre 1)
Mot 10 :  00010 (000A) Nombre d'éléments à lire dans type de mémoire maître
Mot 11 :  00070 (0046) Type de mémoire maître dans lequel les données seront sauvegardées
                       (%I)
Mot 12 :  00001 (0001) Adresse maître dans laquelle les données seront sauvegardées
                       (Entrée 1)
```

Le champ "Nombre d'éléments à lire dans type de mémoire maître" doit être défini en fonction des unités correspondant au type de mémoire de l'équipement maître. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides.

Remarque

L'équipement SNP maître sur le module CMM limite cette commande à 2048 octets maximum (1024 mots).

3.2.10. Commande Ecrire dans mémoire système : 07203 (1C23)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande d'écriture accompagnée de données à laquelle l'équipement esclave répond. Ce service est fourni pour permettre l'accès aux diverses tables de références résidant dans l'API esclave. (Voir dans le Tableau 6.1 les différents types de mémoire supportés). La commande transfère le nombre d'éléments spécifié, de la table de références de l'équipement maître dans celle de l'équipement esclave. Lorsque les types de mémoire des tables de références des équipements maître et esclave(s) diffèrent, la table de références de l'équipement esclave est complétée par autant de 0 que nécessaire.

Exemple de bloc de commande :

Ecrire dans le registre 201 de la mémoire des registres (%R) de l'équipement esclave connecté les entrées 1–10 de la mémoire des entrées (%I) de l'équipement maître. (Les 10 bits de poids faibles du registre %R201 de l'équipement esclave sont remplacés par les données provenant de l'équipement maître ; les bits restants du registre %R201 de l'équipement esclave sont mis à 0).

```
Mot 1 :    00006 (0006) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07203 (1C23) Numéro de la commande SNP

Mot 8 :    00008 (0008) Type de mémoire esclave dans lequel les données seront
                        sauvegardées (%R)
Mot 9 :    00201 (00C9) Adresse esclave dans laquelle les données seront sauvegardées
                        (Registre 201)
Mot 10 :   00010 (000A) Nombre d'éléments à écrire depuis le type de mémoire maître
Mot 11 :   00070 (0046) Type de mémoire maître à partir duquel les données seront
                        écrites (%I)
Mot 12 :   00001 (0001) Adresse maître à partir de laquelle les données seront écrites
                        (Entrée 1)
```

Le champ "Nombre d'éléments à écrire depuis type de mémoire maître" doit être défini en fonction des unités correspondant au type de mémoire de l'équipement maître. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides.

Remarque

L'équipement SNP maître sur le module CMM limite cette commande à 2048 octets maximum (1024 mots).

Remarque

Lorsque les types de mémoire des équipements maître et esclave(s) ne concordent pas, les valeurs sont remplacées par des zéros.

3.2.11. Commande Lire Mémoire des tâches : 07204 (1C24)

Mode : Maître (pour des communications avec un 90–70 esclave uniquement)

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande de lecture à laquelle l'équipement esclave répond par des données. Ce service permet à l'équipement maître de lire le segment de données principal (table de références %P) de la tâche du programme de commande principal de l'équipement esclave. Seuls les API Série 90–70 esclaves supportent ce service ; les API Série 90–20 ou Série 90–30 esclaves généreront des résultats imprévisibles.

Exemple de bloc de commande :

Lire le Mot 1 de la mémoire des tâches du programme de commande principal (%P) de l'équipement esclave connecté et le sauvegarder dans les entrées 1–10 de la mémoire des entrées (%I) de l'équipement maître. Le nom du programme de commande principal de l'équipement esclave est par exemple "MAIN1". Les mots 12–15 renferment le nom du programme principal de l'équipement esclave.

```
Mot 1 :    00009 (0009) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07204 (1C24) Numéro de la commande SNP

Mot 8 :    00001 (0001) Adresse mémoire de la tâche esclave (%P1)
Mot 9 :    00010 (000A) Nombre d'éléments à lire dans type de mémoire maître
Mot 10 :   00070 (0046) Type de mémoire maître dans lequel les données seront
                        sauvegardées (%I)
Mot 11 :   00001 (0001) Adresse maître dans laquelle les données seront sauvegardées
                        (Entrée 1)
Mot 12 :   16717 (414D) Caractères 1 et 2 : M (4Dh, A (41h)
Mot 13 :   20041 (4E49) Caractères 3 et 4 : I (49h), N (4Eh)
Mot 14 :   00049 (0031) Caractères 5 et 6 : 1 (31h), nul

Mot 15 :   00000 (0000) Caractères 7 et 8 : nul, nul
```

Le champ "Nombre d'éléments à lire dans type de mémoire maître" doit être défini en fonction des unités correspondant au type de mémoire de l'équipement maître. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides.

Remarque

L'équipement SNP maître sur le module CMM limite cette commande à 2048 octets maximum (1024 mots).

3.2.12. Commande Ecrire dans mémoire des tâches : 07205 (1C25)

Mode : Maître (pour des communications avec un 90–70 esclave uniquement)

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande d'écriture accompagnée de données à laquelle l'équipement esclave répond. Ce service permet à l'équipement maître d'écrire dans le segment de données principal (table de références %P) de la tâche du programme de commande principal de l'équipement esclave. Lorsque le type de mémoire de la table de références de l'équipement maître ne spécifie pas d'accès au mot, les données de l'équipement esclave sont complétées par autant de zéros que nécessaire. Seuls les API Série 90–70 esclaves supportent ce service ; les API Série 90–20 ou Série 90–30 esclaves généreront des résultats imprévisibles.

Exemple de bloc de commande :

Ecrire dans le Mot de la mémoire des tâches du programme de commande principal (%P) de l'équipement esclave connecté les entrées 1–10 de la mémoire des entrées (%I) de l'équipement maître. Le nom du programme de commande principal de l'équipement esclave est par exemple "MAIN1". Les mots 12–15 renferment le nom du programme principal de l'équipement esclave.

```
Mot 1 :   00009 (0009) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07205 (1C25) Numéro de la commande SNP

Mot 8 :   00001 (0001) Adresse mémoire tâche esclave (%P1)
Mot 9 :   00010 (000A) Nombre d'éléments à écrire depuis type de mémoire maître
Mot 10 :  00070 (0046) Type de mémoire maître à partir duquel les données seront écrites
                       (%I)
Mot 11 :  00001 (0001) Adresse maître à partir de laquelle les données seront écrites
                       (Entrée 1)
Mot 12 :  16717 (414D) Caractères 1 et 2 : M (4Dh, A (41h)
Mot 13 :  20041 (4E49) Caractères 3 et 4 : I (49h), N (4Eh)
Mot 14 :  00049 (0031) Caractères 5 et 6 : 1 (31h), nul

Mot 15 :  00000 (0000) Caractères 7 et 8 : nul, nul
```

Le champ "Nombre d'éléments à écrire depuis type de mémoire maître" doit être défini en fonction des unités correspondant au type de mémoire de l'équipement maître. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides.

Remarque

L'équipement SNP maître sur le module CMM limite cette commande à 2048 octets maximum (1024 mots).

3.2.13. Commande Lire Mémoire des blocs du programme : 07206 (1C26)

Mode : Maître (pour des communications avec un 90–70 esclave uniquement)

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande de lecture à laquelle l'équipement esclave répond par des données. Ce service permet à l'équipement maître de lire le segment de données du sous–bloc local (table de références %L) du bloc de programme spécifié dans l'équipement esclave. Seuls les API Série 90–70 esclaves supportent ce service ; les API Série 90–20 ou Série 90–30 esclaves généreront des résultats imprévisibles.

Exemple de bloc de commande :

Lire les mots 1–10 de la mémoire des blocs du programme (%L) de l'équipement esclave connecté et les sauvegarder dans les registres 201–210 de la mémoire des registres (%R) de l'équipement maître. Le nom du programme de commande principal de l'équipement esclave est "MAIN1" ; le nom du bloc de programme dans l'équipement esclave est "PBLOCK1". Les mots 12–15 renferment le nom du programme principal de l'équipement esclave et les mots 16–19 le nom du bloc de programme de l'équipement esclave.

```
Mot 1 :    00013 (000D) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07206 (1C26) Numéro de la commande SNP

Mot 8 :    00001 (0001) Adresse du bloc de programme (%L1)
Mot 9 :    00010 (000A) Nombre d'éléments à lire dans type de mémoire maître
Mot 10 :   00008 (0008) Type de mémoire maître dans lequel les données seront
                        sauvegardées (%R)
Mot 11 :   00201 (00C9) Adresse maître dans laquelle les données seront sauvegardées
                        (Registre 201)
Mot 12 :   16717 (414D) Caractères 1 et 2 : M (4Dh, A (41h)
Mot 13 :   20041 (4E49) Caractères 3 et 4 : I (49h), N (4Eh)
Mot 14 :   00049 (0031) Caractères 5 et 6 : 1 (31h), nul

Mot 15 :   00000 (0000) Caractères 7 et 8 : nul, nul
Mot 16 :   16976 (4250) Caractères 1 et 2 : P (50h, B (42h)
Mot 17 :   20300 (4F4C) Caractères 3 et 4 : L (4Ch), O (4Fh)
Mot 18 :   19267 (4B43) Caractères 5 et 6 : C (43h), K (4Bh)
Mot 19 :   00049 (0031) Caractères 7 et 8 : 1 (31h), nul
```

Le champ "Nombre d'éléments à lire dans type de mémoire maître" doit être défini en fonction des unités correspondant au type de mémoire de l'équipement maître. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides.

Remarque

L'équipement SNP maître sur le module CMM limite cette commande à 2048 octets maximum (1024 mots).

3.2.14. Commande Ecrire dans mémoire des blocs du programme : 07207 (1C27)

Mode : Maître (pour des communications avec un 90–70 esclave uniquement)

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande d'écriture accompagnée de données à laquelle l'équipement esclave répond. Ce service permet à l'équipement maître d'écrire dans le segment de données du sous–bloc local (table de références %L) du bloc de programme spécifié dans l'équipement esclave. Lorsque le type de mémoire de la table de références de l'équipement maître ne spécifie pas d'accès au mot, les données de l'équipement esclave sont complétées par autant de zéros que nécessaire. Seuls les API Série 90–70 esclaves supportent ce service ; les API Série 90–30 esclaves généreront des résultats imprévisibles.

Exemple de bloc de commande :

Ecrire dans les mots 1–10 de la mémoire des blocs du programme (%L) de l'équipement esclave connecté le contenu des registres 201–210 de la mémoire des registres (%R) de l'équipement maître. Le nom du programme principal de l'équipement esclave est "MAIN1" ; le nom du bloc de programme dans l'équipement esclave est "PBLOCK1". Les mots 12–15 renferment le nom du programme principal de l'équipement esclave et les mots 16–19 le nom du bloc de programme de l'équipement esclave.

```
Mot 1 :    00013 (000D) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07207 (1C27) Numéro de la commande SNP

Mot 8 :    00001 (0001) Adresse mémoire du bloc de programme esclave (%L1)
Mot 9 :    00010 (000A) Nombre d'éléments à écrire depuis équipement maître
Mot 10 :   00008 (0008) Type de mémoire maître à partir duquel les données seront
                        écrites (%R)
Mot 11 :   00201 (00C9) Adresse maître à partir de laquelle les données seront écrites
                        (Registre 201)
Mot 12 :   16717 (414D) Caractères 1 et 2 : M (4Dh, A (41h)
Mot 13 :   20041 (4E49) Caractères 3 et 4 : I (49h), N (4Eh)
Mot 14 :   00049 (0031) Caractères 5 et 6 : 1 (31h), nul

Mot 15 :   00000 (0000) Caractères 7 et 8 : nul, nul
Mot 16 :   16976 (4250) Caractères 1 et 2 : P (50h, B (42h)
Mot 17 :   20300 (4F4C) Caractères 3 et 4 : L (4Ch), O (4Fh)
Mot 18 :   19267 (4B43) Caractères 5 et 6 : C (43h), K (4Bh)
Mot 19 :   00049 (0031) Caractères 7 et 8 : 1 (31h), nul
```

Le champ "Nombre d'éléments à écrire depuis type de mémoire maître" doit être défini en fonction des unités correspondant au type de mémoire de l'équipement maître. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides.

Remarque

L'équipement SNP maître sur le module CMM limite cette commande à 2048 octets maximum (1024 mots).

3.2.15. Commande Etat court de l’API : 07208 (1C28)

Mode : Maître

Description :

Commande déportée. L’équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L’équipement maître envoie une demande Etat court de l’API à laquelle l’équipement esclave répond par des données. Ce service permet à l’équipement maître de lire l’état de l’API esclave.

Exemple de bloc de commande :

Lire l’état court de l’équipement esclave connecté et le sauvegarder dans le registre 201 de la mémoire des registres (%R) de l’équipement maître.

```
Mot 1 :    00003 (0003) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d’état (%R)
Mot 4 :    00000 (0000) Adresse du mot d’état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07208 (1C28) Numéro de la commande SNP

Mot 8 :    00008 (0008) Type de mémoire maître dans lequel l’état sera sauvegardé (%R)
Mot 9 :    00201 (00C9) Adresse maître dans laquelle l’état sera sauvegardé (Registre
                        201)
```

Les champs ”Type et Adresse mémoire maître” doivent être définis de façon à pouvoir contenir les 12 octets de l’état court retournés par l’API esclave. Le tableau ci–dessous décrit la plupart des informations fournies par l’état court de l’API. Elles englobent les 6 octets de données d’état complémentaire fournis à chaque réponse de l’équipement esclave. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides. Voir dans le Tableau 6.12 la description détaillée du mot d’état de l’API esclave.

Le tableau ci–après décrit les principales informations fournies par l'état court de l'API :

Zone de mémorisation de l'état			Description
Mot		Octet	
Mot 1	(octet de poids faible)	Octet 1	Nombre de programmes de commande. 0 = Aucune tâche de programme couramment définie. 1 = Une tâche de programme définie pour la CPU de l'API.
	(octet de poids fort)	Octet 2	Indicateurs de la console de programmation. Ensemble d'indicateurs booléens qui identifient les tâches du programme de commande couramment connectées à des consoles de programmation. Chaque position binaire correspond à une tâche du programme de commande (0 = aucune console de programmation connectée ; 1 = console de programmation connectée), où le bit 0 correspond au programme de commande 0. Etant donné qu'un seul programme de commande est couramment autorisé, cet octet est soit à 1, soit à 0.
Mot 2	(octet de poids faible)	Octet 3	Types d'équipements connectés.
	(octet de poids fort)	Octet 4	Code à 2 bits qui indique le type d'équipement connecté associé à chaque programme de commande. Le champ de 2 bits commençant au bit 0 correspond au programme de commande 0. Etant donné qu'un seul programme de commande est autorisé dans la mise en oeuvre courante, seuls le bit 0 et le bit 1 sont significatifs. Les codes ci–dessous ne s'appliquent qu'aux API Série 90–70. 01h = Console de programmation non spécialisée connectée 00h = Aucune console de programmation connectée.
Mot 3	(octet de poids faible)	Octet 5	Etat auxiliaire. Seul le bit de poids fort est défini et pour les API Série 90–70 uniquement. Le bit 7 indique l'état de la fenêtre de la console de programmation : 0 = Fenêtre de la console de programmation fermée 1 = Fenêtre de la console de programmation ouverte
	(octet de poids fort)	Octet 6	Non utilisé (00h)
Mot 4	(octet de poids faible)	Octet 7	Numéro du programme de commande
	(octet de poids fort)	Octet 8	Niveau de protection courant
Mot 5	(octet de poids faible)	Octet 9	Dernier temps de cycle (en incréments de 0,1 ms)
	(octet de poids fort)	Octet 10	Dernier temps de cycle (en incréments de 0,1 ms)
Mot 6	(octet de poids faible)	Octet 11	Mot d'état de l'API (bits 0–7)
	(octet de poids fort)	Octet 12	Mot d'état de l'API (bits 8–15)

3.2.16. Commande Retourner Nom du programme de commande : 07209 (1C29)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande Retourner Nom du programme de commande à laquelle l'équipement esclave répond par des données. Ce service permet à l'équipement maître d'obtenir de l'équipement esclave le nombre de tâches du programme de commande ainsi que le nom de chacune de ces tâches. Actuellement, un seul programme de commande est supporté par tous les API Série 90. Dans les futures versions, les API Série 90–70 pourront supporter jusqu'à 8 programmes de commande.

Exemple de bloc de commande :

Lire le nom du premier programme de commande dans l'équipement esclave connecté et le sauvegarder dans les registres 201–205 de la mémoire des registres (%R) de l'équipement maître.

```
Mot 1 :    00004 (0004) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07209 (1C29) Numéro de la commande SNP

Mot 8 :    00008 (0008) Type de mémoire maître dans lequel les données du programme de
                        commande seront sauvegardées (%R)
Mot 9 :    00201 (00C9) Adresse maître dans laquelle les données du programme de commande
                        seront sauvegardées (Registre 201)
Mot 10 :   00001 (0001) Nombre de noms de programmes de commande à extraire (1–8)
```

Les champs "Type et Adresse mémoire maître" doivent être définis de façon à pouvoir contenir les données retournées dans la zone Programmes de commande.

Le champ "Nombre de noms de programmes de commande" indique le nombre de noms de programmes de commande esclaves à retourner dans la zone Programmes de commande. La taille de cette zone (en mots) dépend du nombre de noms de programmes de commande demandé et se calcule comme suit : (4* Mot 10) + 1. Les données de la réponse commencent toujours par le premier nom de programme de commande défini dans l'équipement esclave. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides.

Le tableau ci–après décrit le format de la zone Programme de commande retournée par la commande :

Zone Programme de commande	Description
Mot 1	Nombre de programmes de commande
Mot 2	Caractères 1 et 2 du nom de programme de commande 1
Mot 3	Caractères 3 et 4 du nom de programme de commande 1
Mot 4	Caractères 5 et 6 du nom de programme de commande 1
Mot 5	Caractères 7 et 8 du nom de programme de commande 1
.	
.	
.	
Mot 30	Caractères 1 et 2 du nom de programme de commande 8
Mot 31	Caractères 3 et 4 du nom de programme de commande 8
Mot 32	Caractères 5 et 6 du nom de programme de commande 8
Mot 33	Caractères 7 et 8 du nom de programme de commande 8

Le champ ”Nombre de programmes de commande” indique le nombre de programmes de commande définis dans l’équipement esclave.

Le champ ”Nom du programme de commande” renferme le nom ASCII du ou des programmes de commande. Si le nombre de programmes de commande demandé est supérieur au nombre défini, les champs Nom du programme de commande inutilisés renferment la chaîne ASCII nulle (tous les octets à 0).

3.2.17. Commande Retourner Type et ID du contrôleur : 07210 (1C2A)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande Retourner Type et ID de l'API à laquelle l'équipement esclave répond par des données. Ce service permet à l'équipement maître d'obtenir de la CPU de l'API esclave son ID SNP, le type de l'API, le nombre de programmes de commande, le nom du programme, etc.

Exemple de bloc de commande :

Lire le type et l'ID du contrôleur de l'équipement esclave connecté et les sauvegarder dans les registres 201–220 de la mémoire des registres (%R) de l'équipement maître.

```
Mot 1 :    00003 (0003) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07210 (1C2A) Numéro de la commande SNP

Mot 8 :    00008 (0008) Type de mémoire maître dans lequel les informations seront
                        sauvegardées (%R)
Mot 9 :    00201 (00C9) Adresse maître dans laquelle les informations seront
                        sauvegardées (Registre 201)
```

Les champs "Type et Adresse mémoire maître" doivent être définis de façon à pouvoir contenir les 40 octets d'informations retournés par l'API esclave. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides.

Le tableau ci–après décrit le format de la Zone Informations retournée par la commande :

Zone Informations	Description
Mot 1	Caractères 1 et 2 de l'ID SNP de la CPU de l'API
Mot 2	Caractères 3 et 4 de l'ID SNP de la CPU de l'API
Mot 3	Caractères 5 et 6 de l'ID SNP de la CPU de l'API
Mot 4	Caractères 7 et 8 de l'ID SNP de la CPU de l'API
Mot 5	Type majeur/mineur de l'API Série 90 (Octet de poids fort/Octet de poids faible)
Mot 6	Nombre de programmes de commande (Octet de poids faible seulement)
Mot 7	Caractères 1 et 2 du nom du programme de commande principal
Mot 8	Caractères 3 et 4 du nom du programme de commande principal
Mot 9	Caractères 5 et 6 du nom du programme de commande principal
Mot 10	Caractères 7 et 8 du nom du programme de commande principal
Mot 11	Nombre de blocs du programme de commande
Mot 12	Longueur totale des blocs du programme (LSW)
Mot 13	Longueur totale des blocs du programme (MSW)
Mot 14	Somme des totaux de contrôle additifs des blocs du programme
Mot 15	Somme des totaux de contrôle CRC des blocs du programme (LSW)
Mot 16	Somme des totaux de contrôle CRC des blocs du programme (MSW)
Mot 17	Longueur des enregistrements de configuration
Mot 18	Somme des totaux de contrôle additifs des enregistrements de configuration
Mot 19	Somme des totaux de contrôle CRC des enregistrements de configuration (LSW)
Mot 20	Somme des totaux de contrôle CRC des enregistrements de configuration (MSW)

Le tableau ci–après indique les codes de type majeur et mineur de l'API Série 90 :

Type majeur		Description	Type mineur		Description
Dec	Hex		Dec	Hex	
12	0Ch	CPU API Série 90–70	31	1Fh	CPU modèle 731 Série 90–70
			32	20h	CPU modèle 732 Série 90–70
			71	47h	CPU modèle 771 Série 90–70
			72	48h	CPU modèle 772 Série 90–70
			81	51h	CPU modèle 781 Série 90–70
			82	52h	CPU modèle 782 Série 90–70
			88	58h	CPU modèle 788 Série 90–70
			89	59h	CPU modèle 789 Série 90–70
			92	5Ch	CPU modèle 914 Série 90–70
			94	5Eh	CPU modèle 924 Série 90–70
16	10h	CPU API Série 90–20, CPU API Série 90–30	31	1Fh	CPU modèle 211 Série 90–20
			30	1Eh	CPU modèle 311 Série 90–30
			32	20h	CPU modèle 321 Série 90–30
			33	21h	CPU modèle 313 Série 90–30
			34	22h	CPU modèle 323 Série 90–30
			35	23h	CPU modèle 331 Série 90–30
			36	24h	CPU modèle 341 Série 90–30

3.2.18. Commande Retourner Heure/Date de l'API : 07211 (1C2B)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande Retourner Heure/Date de l'API à laquelle l'équipement esclave répond par des données. Ce service permet à l'équipement maître d'obtenir l'heure et la date courantes de l'équipement esclave. Les API Série 90–70 supportent l'heure, la date et le jour de la semaine. L'API Série 90–30 modèle 331 ne supporte pas le jour de la semaine. L'API Série 90–30 modèle 311 et les API Série 90–20 ne supportent ni l'heure, ni la date, ni le jour de la semaine. Les valeurs retournées non supportées n'ont aucune signification et ne doivent pas être utilisées.

Exemple de bloc de commande :

Lire l'heure/la date dans l'équipement esclave connecté et les sauvegarder dans les registres 201–204 de la mémoire des registres (%R) de l'équipement maître.

```
Mot 1 :   00003 (0003) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07211 (1C2B) Numéro de la commande

Mot 8 :   00008 (0008) Type de mémoire maître dans lequel l'heure/la date de l'API
                       seront sauvegardées (%R)
Mot 9 :   00201 (00C9) Adresse maître dans laquelle l'heure/la date de l'API seront
                       sauvegardées (Registre 201)
```

Les champs "Type et Adresse mémoire maître" doivent être définis de façon à pouvoir contenir les 8 octets de données nécessaires pour représenter l'heure/la date retournées par l'API esclave. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides. Toutes les données sont retournées sous un format BCD condensé. Le champ "Jour de la semaine" indique le jour de la semaine où Dimanche = 1 et Samedi = 7. L'heure/la date sont retournées sous le format suivant :

Zone Heure de l'API		Description
Mot	**Octet**	
Mot 1 (octet de poids faible)	Octet 1	Année (en BCD)
Mot 1 (octet de poids fort	Octet 2	Mois (en BCD)
Mot 2 (octet de poids faible)	Octet 3	Jour (en BCD)
Mot 2 (octet de poids fort)	Octet 4	Heure (en BCD)
Mot 3 (octet de poids faible)	Octet 5	Minutes (en BCD)
Mot 3 (octet de poids fort)	Octet 6	Secondes (en BCD)
Mot 4 (octet de poids faible)	Octet 7	Jour de la semaine (Dimanche=1, Samedi=7)
Mot 4 (octet de poids fort)	Octet 8	Non utilisé

3.2.19. Commande Retourner Table des défauts : 07212 (1C2C)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande Retourner Table des défauts à laquelle l'équipement esclave répond par des données. Ce service permet à l'équipement maître d'obtenir tout ou partie des tables de défauts automate ou d'E/S de l'équipement esclave. La table des défauts d'E/S peut comporter jusqu'à 32 entrées et celle des défauts automate jusqu'à 16 entrées.

Exemple de bloc de commande :

Lire l'entrée 1 seulement de la table des défauts d'E/S de l'équipement esclave connecté et la sauvegarder dans les registres 201–227 de la mémoire des registres (%R) de l'équipement maître.

```
Mot 1 :   00006 (0006) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07212 (1C2C) Numéro de la commande

Mot 8 :   00001 (0001) Table des défauts demandée (E/S = 1, API = 2)
Mot 9 :   00001 (0001) Entrée de départ dans table des défauts (E/S = 1–32, API = 1–16)
Mot 10 :  00001 (0001) Nombre d'entrées demandé (E/S = 1–32, API = 1–16)
Mot 11 :  00008 (0008) Type de mémoire maître dans lequel la table des défauts sera
                       sauvegardée (%R)
Mot 12 :  00201 (0001) Adresse maître dans laquelle la table des défauts sera sauvegardée
                       (Registre 201)
```

Les champs "Type et Adresse mémoire maître" doivent être définis de façon à pouvoir contenir les 12 octets de données nécessaires pour représenter l'en–tête de la table plus 42 octets pour chaque entrée de la table des défauts demandée à l'API esclave. Cet espace requis s'applique à la fois aux tables des défauts automate et d'E/S. Voir dans le Tableau 6.1 les différents types et adresses mémoires valides. La zone Table des défauts est retournée sous le format suivant :

Zone Table des défauts	Description
	Date du dernier effacement
Mot 1 (octet de poids faible)	Secondes (en BCD)
Mot 1 (octet de poids fort)	Minutes (en BCD)
Mot 2 (octet de poids faible)	Heures (en BCD)
Mot 2 (octet de poids fort)	Jour (en BCD)
Mot 3 (octet de poids faible)	Mois (en BCD)
Mot 3 (octet de poids fort)	Année (en BCD)
Mot 4	Défauts depuis dernier effacement
Mot 5	Défauts dans la table
Mot 6	Défauts dans cette réponse
Mots 7–27	Entrée #1 de la table des défauts
...	
Mots 322–342	Entrée #16 de la table des défauts
...	
Mots 658–678	Entrée #32 de la table des défauts

Le format de chaque entrée de la table des défauts d'E/S est décrit dans le tableau ci–après :

Zone Table des défauts d'E/S	Description		
Octet 1	Réservé (non utilisé)		
Octets 2–4	Adresse de référence		
Octet 5	Numéro du rack		
Octet 6	Numéro de l'emplacement		
Octet 7	Numéro du bus d'E/S		
Octet 8	Adresse du bus		
Octets 9–10	Adresse du point		
Octet 11	Groupe de défauts		
Octet 12	Action correctrice		
Octet 13	Catégorie du défaut		
Octet 14	Type de défaut		
Octet 15	Description du défaut		
Octets 16–36	Réservés (non utilisés)		
Octet 37	Date :	Secondes	(en BCD)
Octet 38		Minutes	(en BCD)
Octet 39		Heures	(en BCD)
Octet 40		Jour	(en BCD)
Octet 41		Mois	(en BCD)
Octet 42		Année	(en BCD)

Le format de chaque entrée de la table des défauts automate est décrit dans le tableau ci–après :

Zone Table des défauts automate	Description		
Octets 1–4	Réservés (non utilisés)		
Octet 5	Numéro du rack		
Octet 6	Numéro de l'emplacement		
Octet 7	Numéro de l'unité		
Octet 8	Réservé (non utilisé)		
Octet 9	Groupe de défauts		
Octet 10	Action correctrice		
Octets 11–12	Code d'erreur du défaut		
Octets 13–36	Réservés (non utilisés)		
Octet 37	Date :	Secondes	(en BCD)
Octet 38		Minutes	(en BCD)
Octet 39		Heures	(en BCD)
Octet 40		Jour	(en BCD)
Octet 41		Mois	(en BCD)
Octet 42		Année	(en BCD)

3.2.20. Commande Programmer Heure/Date de l'API : 07213 (1C2D)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande Programmer Heure/Date de l'API accompagnée de données à laquelle l'équipement esclave répond. Ce service permet à l'équipement maître de programmer l'heure et la date système dans l'équipement esclave. L'équipement maître dispose de plusieurs options. Il peut spécifier et programmer la date et/ou l'heure de manière explicite. L'équipement maître peut également spécifier que sa propre date/heure système soit utilisée dans l'équipement esclave. Cette option permet de synchoniser la date/l'heure système de l'équipement esclave sur celle de l'équipement maître.

Les API Série 90–70 supportent l'heure, la date et le jour de la semaine. L'API Série 90–30 modèle 331 ne supporte pas le jour de la semaine. L'API Série 90–30 modèle 311 et les API Série 90–20 ne supportent ni l'heure, ni la date, ni le jour de la semaine.

Exemple de bloc de commande :

Programmer la date de l'API Série 90–70 esclave connecté à Dimanche, 19 Janvier 1992.

```
Mot 1 :   00006 (0006) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07213 (1C2D) Numéro de la commande

Mot 8 :   00003 (0003) Définir le mode (heure/date=1, heure seule =2, date seule =3,
                       synchroniser=4)
Mot 9 :   00402 (0192) Année (octet de poids faible)/mois (octet de poids fort)
Mot 10 :  00025 (0019) Jour du mois (octet de poids faible)/heures (octet de poids fort)
Mot 11 :  00000 (0000) Minutes (octet de poids faible)/secondes (octet de poids fort)
Mot 12 :  00001 (0001) Jour de la semaine (octet de poids faible)/0 (octet de poids
                       fort)
```

Le champ "Définir le mode" spécifie le format de date et d'heure à programmer dans l'équipement esclave. Si "Heure seule" est sélectionné, les valeurs correspondant à la date et au jour de la semaine ne sont pas utilisées. Si "Date seule" est sélectionné, la valeur correspondant à l'heure n'est pas utilisée. Si "synchroniser" est sélectionné, l'heure et la date système de l'API maître sont utilisées au lieu des champs Heure et Date du bloc de commande COMREQ.

L'année, le mois, le jour du mois, les heures, les minutes, les secondes et le jour de la semaine doivent être spécifiés sous le format BCD condensé. La valeur Heures doit être entrée sous le format 24 heures (0–23). La valeur Jour de la semaine spécifie le jour de la semaine où Dimanche = 1 et Samedi = 7.

Remarque

Etant donné que les API Série 90–30 ne supportent pas le jour de la semaine, un API Série 90–30 SNP maître ne pourra pas synchroniser un équipement esclave sur le jour de la semaine. Dans ce cas, la valeur 0 doit être envoyée comme Jour de la semaine.

3.2.21. Commande Forcer bascule de la mémoire système : 07214 (1C2E)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande Forcer bascule de la mémoire système accompagnée de données à laquelle l'équipement esclave répond. Ce service permet à l'équipement maître de faire passer un état, un forçage ou un bit de transition de l'équipement esclave à l'état opposé (par exemple, passer de *mettre à 1* à *mettre à 0*, ou passer de *mettre à 0* à *mettre à 1*).

Exemple de bloc de commande :

Faire passer l'Entrée 1 de la mémoire des entrées (%I) de l'équipement esclave connecté à l'état opposé.

```
Mot 1 :   00003 (0003) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07214 (1C2E) Numéro de la commande SNP

Mot 8 :   00070 (0046) Type de mémoire esclave à faire basculer (%I)
Mot 9 :   00001 (0001) Adresse esclave à faire basculer (Entrée 1)
```

Les champs "Type et Adresse mémoire esclave" spécifient le bit de la table de références de l'équipement esclave à faire basculer. Vous ne pouvez spécifier que des types de mémoire orientés Bit. (Voir dans le tableau ci–après les types de mémoire acceptés par cette commande).

Le tableau ci–après indique les types de mémoire de l'équipement esclave que vous pouvez spécifier avec cette commande :

Type de mémoire		Description
Décimal	**Hexadécimal**	
70	46h	Mémoire %I en mode binaire.
72	48h	Mémoire %Q en mode binaire.
74	4Ah	Mémoire %T en mode binaire.
76	4Ch	Mémoire %M en mode binaire.
78	4Eh	Mémoire %SA en mode binaire.
80	50h	Mémoire %SB en mode binaire.
82	52h	Mémoire %SC en mode binaire.
84	54h	Mémoire %S en mode binaire.
86	56h	Mémoire %G en mode binaire.
150	96h	Table de forçages %I en mode binaire.(1)
152	98h	Table de forçages %Q en mode binaire.(1)
154	9Ah	Table de forçages %T en mode binaire.(1)
156	9Ch	Table de forçages %M en mode binaire.(1)
166	A6h	Table de forçages %G en mode binaire.(1)
168	A8h	Table des transitions %I en mode binaire.
170	AAh	Table des transitions %Q en mode binaire.
172	ACh	Table des transitions %T en mode binaire.
174	AEh	Table des transitions %M en mode binaire.
176	B0h	Table des transitions %SA en mode binaire.
178	B2h	Table des transitions %SB en mode binaire.
180	B4h	Table des transitions %SC en mode binaire.
182	B6h	Table des transitions %S en mode binaire.
184	B8h	Table des transitions %G en mode binaire.

1 L'accès à ces tables de références exige le niveau de protection 3. Dans la mesure où ce niveau de protection n'est pas fourni dans les modules CMM Série 90, ces tables de références ne sont pas accessibles.

3.2.22. Commande Etablir un datagramme : 07215 (1C2F)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté pour pouvoir exécuter cette commande ; voir Commande de connexion. L'équipement maître envoie une demande Etablir un datagramme accompagnée de données à laquelle l'équipement esclave répond en renvoyant un ID Datagramme unique. L'équipement maître envoie alors une demande Ecrire un datagramme accompagnée de données supplémentaires à laquelle l'équipement esclave répond. Ce service permet à l'équipement maître de définir un datagramme – zone dans l'équipement esclave où seront regroupées les données de diverses tables de références. Une fois définie, la zone Datagramme peut être accédée via l'ID Datagramme et une commande Mettre à jour le datagramme ou Mettre à jour le datagramme en temps réel.

Exemple de bloc de commande :

Etablir un datagramme permanent dans l'équipement Série 90–70 esclave connecté. Placer l'ID Datagramme fourni par l'équipement esclave dans le registre 161 de la mémoire des registres (%R) de l'équipement maître. Le nom du programme principal de l'équipement esclave est facultatif. Configurer le datagramme pour 2 formats Point. Le format Point 1 renferme les registres 201–210 (%R) de l'équipement esclave et le format Point 2, les entrées 1–16 (%I) de l'équipement esclave. Vous remarquerez que la mémoire des entrées est accédée en mode Octet pour éviter les limites imposées par les API Série 90–70 en mode binaire. La taille des données du datagramme pour tous les formats Point est donc de 10 registres (20 octets) plus 16 entrées (2 octets), soit 22 octets au total.

```
Mot 1 :   00017 (0011) Longueur du bloc de données SNP (doit inclure tous les formats Point)
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07215 (1C2F) Numéro de la commande SNP

Mot 8 :   00000 (0000) Type d'équipement esclave (90-70=0, 90-30/90-20=1)
Mot 9 :   00129 (0081) Type de datagramme (Normal=1, Permanent=81H)
Mot 10 :  00022 (0016) Taille de la zone Datagramme en octets (22 octets)
Mot 11 :  00008 (0008) Type de mémoire maître pour ID Datagramme (%R)
Mot 12 :  00161 (00A1) Adresse maître pour ID Datagramme (Registre 161)
Mot 13 :  00000 (0000) Caractères 1 et 2 du ss-bloc local ou nom du prog principal : nul
Mot 14 :  00000 (0000) Caractères 3 et 4 du ss-bloc local ou nom du prog principal : nul

Mot 15 :  00000 (0000) Caractères 5 et 6 du ss-bloc local ou nom du prog principal : nul
Mot 16 :  00000 (0000) Caractères 7 et 8 du ss-bloc local ou nom du prog principal : nul
Mot 17 :  00002 (0002) Nombre de formats Point à suivre
Mot 18 :  00008 (0008) Type de mémoire (%R) du format Point (1) de l'esclave
Mot 19 :  00201 (00C9) Adresse (%R201) du format Point (1) de l'esclave
Mot 20 :  00010 (000A) Comptage (10 registres : %R201-%R210) du format Point (1) de
                       l'esclave
Mot 21 :  00016 (0010) Type de mémoire (%I en mode Octet) du format Point (2) de l'esclave

Mot 22 :  00001 (0001) Adresse (octet 1 = %I1) du format Point (2) de l'esclave
Mot 23 :  00002 (0002) Comptage (2 octets = 16 entrées : %I1-%I16) du format Point 2 de
                       l'esclave
```

Le champ ”Type d’équipement esclave” indique si l’équipement esclave est dans un API Série 90–70 ou dans un API Série 90–30/90–20. Les résultats sont imprévisibles si cette valeur indique un type d’équipement esclave erroné.

Le champ ”Type de datagramme” définit le datagramme à établir comme un datagramme Normal ou Permanent. Un datagramme normal reste établi dans l’API esclave pendant toute la durée de la communication (exemple, jusqu’à ce qu’une nouvelle demande de connexion SNP ou de mise à jour du datagramme en temps réel soit générée). Un datagramme permanent reste établi dans l’API esclave jusqu’à ce qu’il soit annulé, que l’API soit redémarré ou que le programme de commande du datagramme soit supprimé.

Le champ ”Taille de la zone du datagramme” spécifie la taille de la zone de données du datagramme en octets. Cette zone regroupe toutes les zones de données spécifiées par les formats Point. Chaque format Point définit des données dont la taille varie suivant le type de mémoire et le comptage d’éléments spécifiés. Par exemple, un format Point spécifiant 19 bits de la table des entrées nécessitera 3 octets. Deux registres nécessiteront 4 octets.

Les champs ”Type et Adresse mémoire pour l’ID Datagramme” indiquent l’adresse mémoire de l’API maître dans laquelle cette instruction COMREQ retournera l’ID Datagramme (1 octet). L’ID Datagramme est fourni par l’équipement esclave lorsque la demande Etablir un datagramme a abouti. Cet ID Datagramme sera utilisé par l’équipement maître pour identifier ce datagramme dans toutes les commandes qui suivent. Voir dans le Tableau 1–15 les types et adresses mémoires valides.

Si le datagramme est défini pour accéder à la mémoire des sous–blocs locaux (%L) ou à celle du programme principal (%P) d’un équipement Série 90–70 esclave, le champ ”Ss–bloc local ou Nom du prog principal” doit renfermer le nom du programme principal (pour %P) ou celui du sous–bloc local (pour %L) sous forme de chaîne ASCII. Sinon, ce champ doit être défini à la valeur zéro (0).

Chaque datagramme consiste en un ou plusieurs formats Point. Un format Point définit une gamme d’adresses mémoire dans l’équipement esclave. Le champ ”Nombre de formats Point” indique le nombre de Formats Point à définir. Chaque format Point consiste en 3 mots ; le type de mémoire, l’adresse mémoire et le nombre d’éléments à lire en mémoire. Voir dans le Tableau 6.1, et les Notes ci–après, les types et adresses mémoires valides. Le nombre d’éléments à lire en mémoire est limité entre 1 et 256. Des exemples de formats Point sont illustrés ci–dessus dans les mots 18–20 et 21–23. Un format Point de 3 mots est requis pour chaque format Point compté dans le mot 17. Un maximum de 32 formats Point peuvent être définis dans un datagramme.

Remarque

Il est fortement recommandé de spécifier des types de mémoire orientés Octet dans les formats Point plutôt que des types de mémoire orientés Bit.

Les équipements Série 90–70 esclaves *n'acceptent* dans aucun cas les types de mémoire orientés Bit dans les formats Point d'un datagramme.

Les types de mémoire orientés Bit sont acceptés dans les formats Point du datagramme d'un équipement Série 90–30 esclave, aux restrictions suivantes près : le format Point de type Bit doit entrer dans un seul octet de mémoire aligné. Un octet de mémoire orientée Bit ne renferme que 8 éléments de mémoire (#1 à #8, #9 à #16, ..., #(N*8)+1 à #(N*8)+8, etc.). Le format Point peut commencer sur n'importe quel élément, à la condition expresse que sa longueur n'excède pas l'octet de mémoire renfermant l'élément de départ.

Par exemple, un datagramme Série 90–30 pourra renfermer un format Point de 6 bits contenant les entrées %I33 à %I38, mais pas les entrées %I37 à %I42 ; ce dernier format excédant les limites d'un octet de mémoire aligné (%I33 à %I40).

Remarque

La longueur du bloc de données SNP COMREQ varie suivant le nombre de formats Point définis dans la commande Etablir un datagramme. La longueur du bloc de données COMREQ (mot 1 de l'instruction COMREQ) doit être égale à (3 * Mot 17) + 11.

3.2.23. Commande Mettre à jour le datagramme : 07216 (1C30)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté et un datagramme doit avoir été établi pour pouvoir exécuter cette commande ; voir Commandes de connexion et Etablir un datagramme. L'équipement maître envoie une demande Mettre à jour le datagramme à laquelle l'équipement esclave répond par des données. Ce service permet à l'équipement maître d'extraire une zone de datagramme prédéfinie dans l'équipement esclave. Une fois le datagramme établi, il peut être extrait à partir de son ID autant de fois que nécessaire jusqu'à ce qu'il soit annulé, ou dans le cas d'un datagramme normal, que l'équipement esclave soit déconnecté. Si le datagramme est permanent et que l'équipement esclave est déconnecté, le datagramme n'a pas à être rétabli avant une nouvelle commande Mettre à jour le datagramme.

Exemple de bloc de commande :

Mettre à jour le datagramme permanent défini avec l'ID Datagramme =1 dans l'équipement esclave connecté et sauvegarder les données retournées dans la mémoire des registres (%R) de l'équipement maître, en commençant au Registre 201.

```
Mot 1 :    00005 (0005) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07216 (1C30) Numéro de la commande SNP

Mot 8 :    00001 (0001) ID Datagramme
Mot 9 :    00129 (0081) Type de datagramme (Normal=1, Permanent=81h)
Mot 10 :   00008 (0008) Type de mémoire maître dans lequel le datagramme sera sauvegardé
                        (%R)
Mot 11 :   00201 (00C9) Adresse maître dans laquelle le datagramme sera sauvegardé
                        (Registre 201)
```

Le champ "ID Datagramme" spécifie le datagramme à extraire. C'est la valeur précédemment retournée par le programme d'application API lorsque la demande Etablir un datagramme a abouti.

Le champ "Type de datagramme" indique s'il s'agit d'un datagramme normal ou permanent. La valeur entrée doit correspondre au type spécifié lors de la création du datagramme.

Les champs "Type et Adresse mémoire maître" doivent être définis de façon à pouvoir contenir la zone Datagramme retournée par l'équipement esclave. La taille de cette zone a précédemment été spécifiée dans la commande Etablir un datagramme qui définissait la taille et le format du datagramme. Voir dans le Tableau 6.1 les types et les adresses mémoires valides.

3.2.24. Commande Annuler le datagramme : 07217 (1C31)

Mode : Maître

Description :

Commande déportée. L'équipement esclave doit être connecté et un datagramme doit avoir été établi pour pouvoir exécuter cette commande ; voir Commandes de connexion et Etablir un datagramme. Ce service permet à l'équipement maître d'annuler un datagramme normal ou permanent précédemment établi dans l'équipement esclave.

Exemple de bloc de commande :

Annuler le datagramme permanent dont l'ID Datagramme =1 dans l'équipement esclave connecté.

```
Mot 1 :   00003 (0003) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07217 (1C31) Numéro de la commande SNP

Mot 8 :   00001 (0001) ID Datagramme
Mot 9 :   00129 (0081) Type de datagramme (Normal=1, Permanent=81h)
```

Le champ "ID Datagramme" spécifie le datagramme à annuler. C'est la valeur précédemment retournée par le programme d'application API lorsque la demande Etablir un datagramme a abouti. La valeur spéciale –1 indique que tous les datagrammes du type sélectionné seront annulés.

Le champ "Type de datagramme" indique s'il s'agit d'un datagramme normal ou permanent. La valeur entrée doit correspondre au type spécifié lors de la création du datagramme.

Remarque

Si l'équipement esclave est un module CMM311 (Série 90–30) dont les deux ports sont configurés en mode SNP esclaves, la commande Annuler le datagramme avec l'ID Datagramme spéciale –1 annulera tous les datagrammes du type spécifié sur *les deux* ports.

3.2.25. Commande Mettre à jour le datagramme en temps réel : 07218 (1C32)

Mode : Maître

Description :

Commande déportée. Un datagramme permanent doit avoir été établi pour pouvoir exécuter cette commande. Ce service permet à l'équipement maître de se connecter à l'équipement esclave et d'extraire le datagramme permanent spécifié à l'aide d'une seule commande. Ce service diffère du service Mettre à jour le datagramme dans le sens où aucune communication avec l'équipement esclave n'a besoin d'être préalablement établie à l'aide d'une commande de connexion ou de connexion longue. Les commandes Mettre à jour le datagramme en temps réel peuvent être générées en fonction des besoins ; aucune commande de connexion préalable pour établir la communication avec les équipements esclaves désirés n'est requise. Ce service est fourni pour permettre d'extraire les données le plus rapidement possible d'un équipement esclave, notamment dans une application de scrutation en mode multipoint. Cette commande ne peut toutefois être utilisée que pour extraire des datagrammes permanents.

Exemple de bloc de commande :

Mettre à jour (à l'aide du service Mettre à jour le datagramme en temps réel) le datagramme permanent dont l'ID Datagramme =1 dans l'équipement esclave dont l'ID SNP est "SNP ID1". Sauvegarder le datagramme retourné dans la mémoire des registres (%R) de l'équipement maîte, en commençant au registre 201. Sauvegarder les 6 octets de l'état complémentaire dans les registres 171–173 de la mémoire des registres (%R) de l'équipement maître.

```
Mot 1 :    00010 (000A) Longueur du bloc de données SNP
Mot 2 :    00000 (0000) Mode NOWAIT
Mot 3 :    00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :    00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :    00000 (0000) Non utilisé
Mot 6 :    00000 (0000) Non utilisé
Mot 7 :    07218 (1C32) Numéro de la commande SNP

Mot 8 :    00001 (0001) ID Datagramme
Mot 9 :    20051 (4E53) Caractères 1 et 2 de l'ID SNP esclave : S (53h), N (4Eh)
Mot 10 :   18768 (4950) Caractères 3 et 4 de l'ID SNP esclave : P (50h), I (49h)
Mot 11 :   12612 (3144) Caractères 5 et 6 de l'ID SNP esclave : D (44h), 1 (31h)
Mot 12 :   00000 (0000) Caractères 7 et 8 de l'ID SNP esclave : nul, nul
Mot 13 :   00008 (0008) Type de mémoire maître dans lequel le datagramme sera sauvegardé
                        (%R)
Mot 14 :   00201 (00C9) Adresse maître dans laquelle le datagramme sera sauvegardé
                        (Registre 201)
Mot 15 :   00008 (0008) Type de mémoire maître dans lequel l'état complémentaire sera
                        sauvegardé (%R)
Mot 16 :   00171 (00AB) Adresse maître dans laquelle l'état complémentaire sera
                        sauvegardé (Registre 171)
```

Le champ "ID Datagramme" spécifie le datagramme permanent à extraire. C'est la valeur précédemment retournée par le programme d'application API lorsque la demande Etablir un datagramme a abouti.

Le champ "ID SNP esclave" spécifie l'ID SNP de l'équipement esclave désiré. Ce champ comporte 8 octets. Pour les API Série 90–70 esclaves, l'ID SNP peut comporter 7 octets maximum suivis d'un caractère nul (0) et accepte tous les caractères ASCII. Pour les API Série 90–30 ou Série 90–20 esclaves, l'ID SNP est limité à 6 octets maximum suivis d'un caractère nul (0) ; les 6 octets ne peuvent renfermer que les caractères ASCII '0' à '9' compris et les majuscules 'A' à 'F' comprises.

L'ID SNP nul (caractère 1 = 0) peut être utilisé pour sélectionner n'importe quel équipement SNP esclave, indépendamment de l'ID SNP qui lui a été affecté. Les commandes avec des ID SNP nuls ne peuvent aboutir *que* dans

une configuration point–à–point. Dans une configuration multipoint, tous les équipements SNP esclaves répondront à une commande de connexion longue spécifiant un ID SNP nul. Cette procédure donne toutefois des résultats imprévisibles et ne doit pas être tentée.

Les champs ”Type et Adresse mémoire maître” doivent être définis de façon à pouvoir contenir la zone Datagramme retournée par l’équipement esclave. La taille de cette zone a été précédemment spécifiée dans la commande Etablir un datagramme qui définissait la taille et le format du datagramme. Voir dans le Tableau 6.1 les types et les adresses mémoires valides.

Les champs Type et Adresse mémoire maître dans lequel/laquelle l’état complémentaire sera sauvegardé sont optionnels. S’ils sont définis, la zone qui renfermera les 6 octets de l’état complémentaire doit également être définie. Les données relatives à l’état complémentaire sont retournées avec chaque message de réponse SNP transmis par l’équipement esclave. Une fois la zone d’état complémentaire spécifiée, c’est là que seront enregistrées toutes les données relatives à l’état complémentaire chaque fois qu’un message de réponse SNP Connexion réussie sera reçu et ce, tant que la connexion reste établie. Si le Type et l’Adresse mémoire maître dans lequel/laquelle l’état complémentaire sera sauvegardé sont définis à zéro (aucune zone complémentaire n’est spécifiée), les données relatives à l’état complémentaire sont ignorées et ne sont pas fournies à l’application en diagramme en échelle de l’API. (Voir dans le Tableau 6.11 la description des données d’état complémentaires). Voir dans le Tableau 6.1 les types et adresses mémoires valides.

Remarque

La commande Mettre à jour le datagramme en temps réel peut être envoyée à tout équipement esclave, spécifié par l’ID SNP. Cette commande génère une séquence Break, qui annulera toutes les commandes précédentes de connexion ou connexion longue et arrêtera prématurément toute communication SNP existante. Une nouvelle commande de connexion ou connexion longue doit être générée pour rétablir les communications SNP normales, une fois que toutes les commandes Mettre à jour le datagramme en temps réel ont abouti.

3.2.26. Commande Connexion longue : 07300 (1C84)

Mode : Maître

Description :

Commande déportée. La commande Connexion longue permet de spécifier de nombreux paramètres supplémentaires pour améliorer les communications entre l'équipement maître et l'équipement esclave. Cette commande peut être générée à tout instant. Après quoi, une séquence Break est envoyée à tous les équipements esclaves. La détection d'une séquence Break par un équipement esclave arrête immédiatement toute session de communication en cours et place l'équipement esclave en état d'attente d'une demande de connexion valide. L'équipement maître attend que le laps de temps défini par le temporisateur T4 ait expiré pour envoyer la demande de connexion aux équipements esclaves. Il attend ensuite une réponse de connexion. Seul l'équipement esclave dont l'ID SNP correspond répondra. Les autres reviendront automatiquement à l'état d'attente d'une séquence Break.

Outre l'échange demande/réponse de connexion, une demande/réponse Sélectionner les paramètres SNP est également échangée. La demande Sélectionner les paramètres SNP est générée par l'équipement SNP maître pour négocier avec l'équipement SNP esclave la valeur des temporisateurs T2 et T3', le délai de retournement du modem, le délai de transmission et la taille maximale des données SNP.

Une fois connecté à un API esclave, l'équipement maître bénéficie automatiquement du niveau de protection défini par défaut dans l'équipement esclave. Pour le port interne de la CPU d'un API Série 90–70, le niveau de protection par défaut des équipements esclaves est le Niveau 0 ; le niveau 0 protège la mémoire de l'API en lecture et en écriture. Pour le port interne de la CPU d'un API Série 90–30 ou Série 90–20, le niveau de protection par défaut des équipements esclaves est le Niveau 1 ; le niveau 1 autorise la lecture de la mémoire de l'API mais interdit tout accès en écriture. Pour un équipement SNP esclave sur le module CMM d'un API Série 90, le niveau de protection est toujours le Niveau 2 et il ne peut pas être modifié ; le niveau 2 autorise les accès en lecture et en écriture dans la mémoire de l'API.

Si l'équipement maître requiert d'autres niveaux de protection, voir commande Modifier le niveau de protection, dans le présent chapitre.

La commande de connexion longue valide ou invalide également la signalisation de l'état complémentaire pendant la durée de la communication SNP. Lorsqu'il est validé, l'état complémentaire est mis à jour après chaque commande ayant abouti.

Exemple de bloc de commande :

Générer une commande Connexion longue sur l'équipement esclave dont l'ID SNP est "SNP ID1". Sauvegarder les 6 octets de l'état complémentaire dans les registres 171–173 de la mémoire des registres (%R) de l'équipement maître. Désactiver le temporisateur T3' pour éliminer l'échange du message "keep alive". Faire passer le temporisateur T4 à 100 ms. Programmer tous les autres temporisateurs SNP sur leur valeur de configuration par défaut. Programmer le délai de retournement du modem sur 10 ms pour les transmissions par modem et réduire la taille maximale du message SNP à 256 octets pour les transmissions en environnements bruyants.

```
Mot 1 :   00015 (000F) Longueur du bloc de données SNP
Mot 2 :   00000 (0000) Mode NOWAIT
Mot 3 :   00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :   00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :   00000 (0000) Non utilisé
Mot 6 :   00000 (0000) Non utilisé
Mot 7 :   07300 (1C84) Numéro de la commande SNP

Mot 8 :   20051 (4E53) Caractères 1 et 2 de l'ID SNP esclave : S (53h), N (4Eh)
Mot 9 :   18768 (4950) Caractères 3 et 4 de l'ID SNP esclave : P (50h), I (49h)
Mot 10 :  12612 (3144) Caractères 5 et 6 de l'ID SNP esclave : D (44h), 1 (31h)
Mot 11 :  00000 (0000) Caractères 7 et 8 de l'ID SNP esclave : nul, nul
Mot 12 :  00008 (0008) Type de mémoire maître dans lequel l'état complémentaire sera
                       sauvegardé (%R)
Mot 13 :  00171 (00AB) Adresse maître dans laquelle l'état complémentaire sera sauvegardé
                       (Registre 171)
Mot 14 :  00005 (0005) T1 (5 ms ; défaut = 5 ms)

Mot 15 :  02000 (07D0) T2 (0 = désactivé) (2 s ; défaut = 2 s)
Mot 16 :  00000 (0000) T3' (0 = désactivé ; défaut = 10 s)
Mot 17 :  00100 (0064) T4 (100 ms ; défaut = 50 ms)
Mot 18 :  00000 (0000) T5 (0 = désactivé ; défaut = désactivé)
Mot 19 :  00010 (000A) Délai de retournement du modem (10 ms ; défaut = 0 ms)
Mot 20 :  00000 (0000) Délai de transmission (0 ms ; défaut = 0 ms)
Mot 21 :  00256 (0100) Taille maximale des données SNP (256 octets ; défaut = 1000 octets)
```

Le champ "ID SNP esclave" spécifie l'ID SNP de l'équipement esclave désiré. Ce champ comporte 8 octets. Pour les API Série 90–70 esclaves, l'ID SNP peut comporter 7 octets maximum suivis d'un caractère nul (0) et accepte tous les caractères ASCII. Pour les API Série 90–30 ou Série 90–20 esclaves, l'ID SNP est limité à 6 octets maximum suivis d'un caractère nul (0) ; les 6 octets ne peuvent renfermer que les caractères ASCII '0' à '9' compris et les majuscules 'A' à 'F' comprises.

L'ID SNP nul (caractère 1 = 0) peut être utilisé pour sélectionner n'importe quel équipement SNP esclave, indépendamment de l'ID SNP qui lui a été affecté. Les commandes avec des ID SNP nuls ne peuvent aboutir *que* dans une configuration point–à–point. Dans une configuration multipoint, tous les équipements SNP esclaves répondront à une commande de connexion longue spécifiant un ID SNP nul. Cette procédure donne toutefois des résultats imprévisibles et ne doit pas être tentée.

Les champs Type et Adresse mémoire maître dans lequel/laquelle l'état complémentaire sera sauvegardé sont optionnels. S'ils sont définis, la zone qui renfermera les 6 octets de l'état complémentaire doit également être définie. Les données relatives à l'état complémentaire sont retournées avec chaque message de réponse SNP transmis par l'équipement esclave. Une fois la zone d'état complémentaire spécifiée, c'est là que seront enregistrées toutes les données relatives à l'état complémentaire chaque fois qu'un message de réponse SNP Connexion réussie sera reçu et ce, tant que la connexion reste établie. Si le Type et l'Adresse mémoire maître dans lequel/laquelle l'état complémentaire sera sauvegardé sont définis à zéro (aucune zone complémentaire n'est spécifiée), les données relatives à l'état complémentaire sont ignorées et ne sont pas fournies à l'application en diagramme en échelle de l'API. (Voir dans le Tableau 6.11 la description des données d'état complémentaires). Voir dans le Tableau 6.1 les types et adresses mémoires valides.

Les champs "T1", "T2", "T3', "T4", et "T5" spécifient les nouvelles valeurs des différents temporisateurs SNP. La valeur de tous les temporisateurs SNP doit être spécifiée en millisecondes. La valeur des temporisateurs T1, T2, et T3' sera négociée entre l'équipement maître et les équipements esclaves. (Voir le Chapitre 7, Définition des protocoles, § 3, Protocole SNP, pour plus d'informations sur les temporisateurs SNP et leur utilisation). Dans cette instruction COMREQ, la valeur 0 désactive le temporisateur concerné.

Lorsqu'un modem est utilisé, le temporisateur T4 doit être défini à 600 ms ou plus pour que la séquence Break soit transmise correctement.

Le champ "Délai de retournement du modem" spécifie le laps de temps requis par le modem pour retourner la liaison. Ce délai se spécifie en millisecondes et sa valeur par défaut est 0.

Le champ "Délai de transmission" spécifie un laps de temps en secondes qui tient compte des délais de transmission inhabituellement longs entre des équipements SNP. Les délais inhabituellement longs sont typiques des liaisons par satellite. Ce délai se spécifie en secondes et sa valeur par défaut est 0.

Le champ "Taille maximale des données SNP" permet de réduire la taille des messages de données SNP transmis par le câble. La taille des messages gagne à être réduite dans les environnements bruyants pour minimiser les erreurs de transmission. La taille des messages peut être définie entre 42 et 1000 octets et doit correspondre à un nombre pair d'octets. Sa valeur par défaut est 1000.

3.2.27. Commande Numérotation automatique : 07400 (1CE8)

Mode : Maître

Description :

Commande déportée. Cette commande permet de transmettre une chaîne de données à un modem externe raccordé au port série ; elle permet de numéroter automatiquement le modem connecté à l'aide de commandes spéciales pour modem. Ces chaînes de commande respectent la convention Hayes. Elles ne font pas partie du protocole SNP.

Lorsque la commande Numérotation automatique est générée, une séquence d'échappement est transmise au modem connecté. La séquence d'échappement assure que le modem revient à l'état de la commande. Elle consiste en 2 secondes de silence, suivies de 3 caractères plus ("+") et de 2 autres secondes de silence. La chaîne de commande spécifiée dans le COMREQ Numérotation automatique sera alors transmise immédiatement après la séquence d'échappement. Son format dépend du modem connecté et de l'opération désirée. Une fois la liaison téléphonique établie, c'est le programme d'application API qui raccrochera en générant une nouvelle commande Numérotation automatique et en envoyant la chaîne de commande de raccrochage appropriée.

Pour être reconnue par le modem, chaque chaîne de commande doit se terminer par le caractère ASCII Retour chariot (0Dh). Ce caractère doit être inclus dans la chaîne de commande et compté dans la longueur.

La Longueur du bloc de données SNP doit inclure toute la chaîne de commande. Sa taille dépend par conséquent de celle de la chaîne de commande. La Longueur du bloc de données COMREQ (mot 1 de l'instruction COMREQ) doit être égale à ((Mot 9 + 1) / 2) + 3.

Exemple de bloc de commande :

Composer le numéro 234–5678. La chaîne de commande utilisée est "ATDT2345678<CR>". Le temps de réponse maximal du modem est 30 secondes. (Il sera vérifié). Les données de la chaîne de commande destinée au modem commence au mot 10.

```
Mot 1 :     00009 (0009) Longueur du bloc de données SNP
                         (chaîne de commande comprise)
Mot 2 :     00000 (0000) Mode NOWAIT
Mot 3 :     00008 (0008) Type de mémoire du mot d'état (%R)
Mot 4 :     00000 (0000) Adresse du mot d'état moins 1 (Registre 1)
Mot 5 :     00000 (0000) Non utilisé
Mot 6 :     00000 (0000) Non utilisé
Mot 7 :     07400 (1CE8) Numéro de la commande SNP

Mot 8 :     00030 (001E) Temps limite de réponse du modem (30 secondes)
Mot 9 :     00012 (000C) Nombre d'octets dans la chaîne de commande
Mot 10 :    21569 (5441) A (41h), T (54h)
Mot 11 :    21572 (5444) D (44h), T (54h)
Mot 12 :    13106 (3332) 2 (32h), 3 (33h)
Mot 13 :    13620 (3534) 4 (34h), 5 (35h)
Mot 14 :    14134 (3736) 6 (36h), 7 (37h)

Mot 15 :    03384 (0D38) 8 (38h), <CR>(0Dh)
```

Le champ ”Temps limite de la réponse du modem” spécifie le laps de temps maximal pendant lequel l’instruction COMREQ attendra la chaîne de réponse complète du modem, après que la commande a été envoyé au modem. Ce temps limite doit être spécifié en secondes. S’il est défini à zéro, le temps de réponse du modem ne sera pas vérifié ; l’instruction COMREQ aboutit dès que la chaîne de commande est envoyée au modem. S’il est défini à une valeur non nulle, le temps de réponse du modem est vérifié ; l’instruction COMREQ attend la réponse du modem. Dès que la réponse est reçue, elle est vérifiée pour s’assurer que la commande envoyée au modem a abouti ; si la réponse n’arrive pas dans le temps imparti, un code d’erreur est généré.

Le champ ”Nombre d’octets dans la chaîne de commande” spécifie la longueur de la chaîne de commande qui sera transmise au modem. Tous les caractères doivent être comptés, espaces compris, ainsi que le caractère ASCII Retour chariot de fin <CR>. La chaîne de commande du modem peut comporter de 1 à 250 octets.

Des exemples de chaînes de commande les plus courantes envoyées à des modems compatibles Hayes sont fournis ci–après :

Chaîne de commande	Longueur	Fonction
ATDP15035559999<CR>	16 (10h)	Appel par impulsions du numéro 1–(503) 555–9999
ATDT15035559999<CR>	16 (10h)	Appel par boutons–poussoirs du numéro 1–(503) 555–9999
ATDT9,15035559999<CR>	18 (12h)	Appel par boutons–poussoirs utilisant une ligne extérieure avec pause
ATH0<CR>	5 (05h)	Raccrochage du téléphone
ATZ<CR>	4 (04h)	Retour de la configuration du modem à des valeurs enregistréesen interne.

Page laissée blanche intentionnellement

Chapitre 7 *Définition des protocoles – CCM, RTU, SNP et SNP–X*

Ce chapitre traite des sujets suivants :

- 1 : Protocole CCM
- 2 : Protocole RTU
- 3 : Protocole SNP
- 4 : Protocole SNP–X

Chaque paragraphe donne une description complète du protocole, tel qu'il est mis en oeuvre dans le module CMM.

1. PROTOCOLE CCM

Le protocole CCM est décrit dans ses moindres détails pour les utilisateurs désireux de développer un gestionnaire de communication série CCM. Au nombre des sujets traités figurent le flux du protocole en modes égal à égal et maître–esclave, ainsi que les spécifications de synchronisation et du nombre d'essais. Des tables d'états sont fournies à la fin du paragraphe pour illustrer ce qui se passe lors des échanges du protocole en modes égal à égal et maître–esclave. Des tables distinctes sont fournies pour les CCM d'égal à d'égal, CCM maître et CCM esclave.

Le protocole CCM transfère les données en utilisant le format de données série asynchrone. Pour plus d'informations sur la transmission série et le format asynchrone, voir Chapitre 2, *Série 90 : notions de communication série*, § 2, *L'interface série*. Le présent paragraphe traite des sujets suivants :

- Présentation du protocole CCM
- Description du mode CCM égal à égal
- Description du mode CCM maître–esclave
- Spécifications de synchronisation et du nombre d'essais
- Tables d'états du protocole CCM

1.1. PRÉSENTATION DU PROTOCOLE CCM

Le protocole CCM se base sur la norme ANSI X3.28 pour transmettre des caractères asynchrones sous un format binaire 8 bits ou le format ASCII avec bit de parité en option. La parité peut être définie à Impaire ou Sans.

1.1.1. Modes de fonctionnement

Le protocole CCM offre deux modes de fonctionnement : égal à égal et maître–esclave.

Le mode égal à égal est une configuration point–à–point dans laquelle seulement deux équipements partagent la même ligne de communication. L'un ou l'autre équipement peut établir la communication. L'équipement qui établit la communication est appelé Equipement source et celui qui répond, l'Equipement destinataire.

Le mode maître–esclave est typique d'une configuration multipoint avec un seul équipement maître et un ou plusieurs équipements esclaves. Seul l'équipement maître peut établir les communications.

1.1.1.1. Caractères de commande CCM

Le protocole CCM utilise des caractères de commande ASCII spéciaux tant en mode égal à égal qu'en mode maître–esclave. Le tableau ci–après indique les caractères de commande ASCII utilisés par le protocole CCM :

Tableau 7-1. Caractères de commande utilisés par le protocole CCM

	Valeur hex	Signification
ENQ	05	Demande de renseignements
ACK	06	Accusé de réception
NAK	15	Accusé de réception négatif
SOH	01	Début d'en–tête
ETB	17	Fin de bloc
STX	02	Début de texte
ETX	03	Fin de texte
EOT	04	Fin de transmission

1.1.1.2. Flux du protocole CCM

Un exemple d'échange de messages sous protocole CCM est fourni ci–après. Le flux du message est identique pour les modes égal à égal et maître–esclave ; seuls les caractères transmis lors de la séquence ENQ diffèrent légèrement. Ces différences seront expliquées plus loin dans ce paragraphe.

Séquence Demande de renseignements (ENQ). La séquence ENQ est une demande d'établissement de la communication qui est transmise par un équipement maître ou homologue (mode égal à égal). Si l'équipement destinataire n'est pas occupé, il envoie un accusé de réception positif (ACK). S'il est occupé, il envoie un accusé de réception négatif (NAK).

Transfert de l'en–tête. Une fois qu'il a reçu un accusé de réception positif en réponse à sa séquence ENQ, l'équipement initiateur envoie un message de 17 caractères appelé En–tête. Le bloc d'en–tête spécifie le sens, le type et l'adresse mémoire, ainsi que la longueur du transfert de données. Tous les octets du bloc d'en–tête, exception faite des caractères de commande CCM, sont au format hexadécimal codé ASCII.

Si l'en–tête est correct, la station destinataire répondra par un accusé de réception positif (ACK). S'il est incorrect, elle répondra par un accusé de réception négatif (NAK).

Le format de l'en–tête et les différents champs qu'il contient sont illustrés ci–dessous :

SOH	ID Destinataire	Sens flux des données & Mémoire Destinataire	Type de mémoire Destinataire	Adresse mémoire Destinataire Octet PF	Adresse mémoire Destinataire Octet Pf	Nb de blocs de données complets	Nb d'octets dans dernier bloc	ID Source	ETB	LRC
1	2 3	4	5	6 7	8 9	10 11	12 13	14 15	16	17

Tableau 7-2. Description du bloc d'en–tête

Champ	Octet	Description	Gamme valide
SOH	1	Caractère de commande Début d'en–tête	01h
ID de l'équipement destinataire	2,3	Numéro d'identification de l'équipement destinataire. Valeur hexadécimale codée en ASCII. En mode égal à égal, la valeur FFh (255 en décimal) est reconnue et acceptée par tous les équipements.	1–255 en décimal (égal à égal) 1–90 en décimal (maître–esclave)
Sens du flux des données/ Type de mémoire du destinataire	4,5	Les octets 4 et 5 fournissent le type de mémoire de l'équipement destinataire. L'octet 4 spécifie également le sens du transfert (lecture ou écriture). Valeurhexadécimale codée en ASCII.	Dépend de l'équipement destinataire. Voir Tableau 7.3
Adresse mémoire du destinataire	6–9	Ce champ spécifie l'adresse de départ pour le transfert de données. Valeur hexadécimale codée en ASCII.	Dépend de l'équipement destinaire. Voir Tableau 5.5
Nombre de blocs de données complets	10,11	Nombre de blocs de données complets (256 octets) à transférer. Valeur hexadécimale codée en ASCII.	0–255 en décimal
Nombre d'octets du dernier bloc	12,13	Nombre d'octets (moins de 256) contenus dans le dernier ou seul bloc de données à transférer. Valeurhexadécimale codée en ASCII.	0–255 en décimal
ID Source	14,15	Numéro d'identification de l'équipement source. Valeur hexadécimale codée en ASCII.	1–255 en décimal (égal à égal) 1–90 en décimal (maître–esclave)
ETB	16	Caractère de commande Fin de bloc	17h
LRC	17	Contrôle de redondance longitudinale LRC. OU exclusif (XOR) des octets 2 à 15.	n/a

Exemple de valeur hexadécimale codée en ASCII

La plupart des valeurs sont spécifiées en hexadécimal codé ASCII. Chaque chiffre hexadécimal est converti en un caractère ASCII. Si par exemple l'adresse mémoire de l'équipement destinataire est 00986 (03DAh), chaque chiffre hexadécimal sera converti en un caractère ASCII, comme illustré ci–après.

Octet	6	7	8	9
Adresse Mem. Dest. (hexadécimal)	0	3	D	A
Adresse Mem. Dest. (hex. codé ASCII)	30	33	44	41

Tableau 7-3. Types de mémoire Destinataire pour services CCM

Type mémoire Destinataire (Décimal)	Octet 4[2]		Octet 5		Type de mémoire Destinataire
	(Hex) Lr/Ec	(ASCII) Lr/Ec	(Hex) Lr/Ec	(ASCII) Lr/Ec	
0	30/38	0/8	30	0	Adresse mémoire absolue de la CPU
1	30/38	0/8	31	1	Table des registres de la CPU (1)
2	30/38	0/8	32	2	Table des entrées de la CPU (1)
3	30/38	0/8	33	3	Table des sorties de la CPU (1)
4	30/38	0/8	34	4	Table des entrées forcées de la CPU
5	30/38	0/8	35	5	Table des sorties forcées de la CPU
6	30/38	0/8	36	6	Zone de travail de la CPU (1)
7	30/38	0/8	37	7	Programme d'application de la CPU
8	30/38	0/8	38	8	Tampon Accès rapide CCM
9	30/38	0/8	39	9	Mots d'état de diagnostic CCM (1)
13	38	8	44	D	Mettre à 1 bits de la table des entrées (1)
14	38	8	45	E	Mettre à 1 bits de la table des sorties (1)
15	38	8	46	F	Mettre à 1 bits de la table des entrées forcées
16	39	9	30	0	Mettre à 1 bits de la table des sorties forcées
17	39	9	31	1	Mettre à 0 bits de la table des entrées (1)
18	39	9	32	2	Mettre à 0 bits de la table des sorties (1)
19	39	9	33	3	Mettre à 0 bits de la table des entrées forcées
20	39	9	34	4	Mettre à 0 bits de la table des sorties forcées
21	39	9	35	5	Faire basculer bits de la table des entrées
22	39	9	36	6	Faire basculer bits de la table des sorties

[1] Types de mémoire valides pour API Série 90.
[2] Les fonctions binaires correspondent à des demandes d'écriture seulement.

Exemple d'en–tête CCM. Dans l'exemple qui suit, l'équipement source (ID = 02) lit 10 registres à partir du registre %R00986 dans l'équipement destinataire dont l'ID destinataire = 01.

Octet	Champ	Binaire	Hex	ASCII
1	Début d'en–tête	0000 0001	01	SOH
2	ID destinataire – MSB	0011 0000	30	0
3	ID destinataire – LSB	0011 0001	31	1
4	Sens des données (Lecture)	0011 0000	30	0
5	Type Mémoire dest. (Registre)	0011 0001	31	1
6	Adresse Mémoire dest. – PF	0000 0000	30	0
7	Adresse Mémoire dest.– NPF	0011 0011	33	3
8	Adresse Mémoire dest.– NPF	0100 0100	44	D
9	Adresse Mémoire dest.– Pf	0100 0001	41	A
10	Nb de blocs complets – PF	0011 0000	30	0
11	Nb de blocs complets – Pf	0011 0000	30	0
12	Nb d'octets dernier bloc – PF	0011 0001	31	1
13	Nb d'octets dernier bloc – Pf	0011 0100	34	4
14	ID Source – PF	0011 0000	30	0
15	ID Source – Pf	0011 0010	32	2
16	Bloc Fin de transfert	0001 0111	17	ETB
17	Caractère de contrôle de bloc (LRC)	0000 0001	01	–

* La valeur LRC est le résultat du XOR vertical (OU exclusif) des octets 2 – 15. Les nombres similaires s'annulent mutuellement à zéro.

Transfert des données. Le transfert commence dès réception d'un accusé de réception positif (ACK) de la séquence d'en–tête. Les données sont écrites vers ou lues depuis le répondeur, suivant la valeur spécifiée pour le bit de Sens des données dans l'en–tête.

Si la taille en octets des données à transférer est inférieure ou égale à 256, le transfert de données n'utilisera qu'un seul bloc de données. Si elle est supérieure à 256, le transfert utilisera autant de blocs que nécessaire.

Le format du bloc de données et les champs qu'il renferme sont illustrés ci–après :

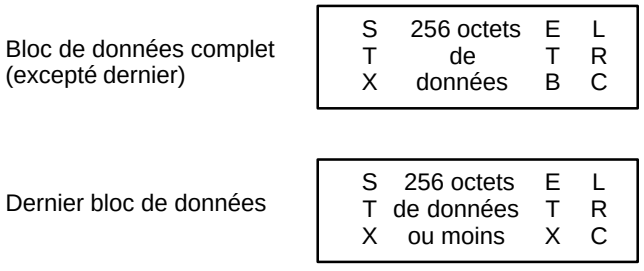


Figure 7-1. Format du bloc de données

Tableau 7-4. Description du bloc de données

Champ	Description	Gamme valide
STX	Caractère de commande Début de texte	02h
Données	1–256 caractères.	–
ETB/ETX	Caractère de commande Fin de bloc utilisé si le bloc de données n'est pas le dernier.	17h
	Caractère de commande Fin de bloc texte utilisé si c'est le dernier bloc de données.	03h
LRC	Contrôle de redondance longitudinale sur tous les caractères de données du bloc.	–

Arrêt du protocole. Une fois le dernier bloc de données transféré avec succès, un caractère de commande EOT est utilisé pour désactiver la connexion entre des équipements homologues (mode égal à égal) ou entre l'équipement maître et le(s) équipement(s) esclave(s).

En mode égal à égal, tout comme en mode maître–esclave, la station qui envoie le dernier bloc de données envoie le caractère EOT dès réception de l'ACK du dernier bloc de données. De plus, en mode maître–esclave, l'équipement maître envoie toujours un EOT pour mettre fin à une transaction.

Le caractère de commande EOT est également utilisé pour arrêter prématurément une conversation si une partie du protocole CCM ne parvient pas à s'exécuter correctement.

1.2. MODE CCM D'ÉGAL À ÉGAL

Ce paragraphe décrit le flux du protocole ainsi que la procédure de détection/correction des collisions en mode CCM d'égal à égal.

1.2.1. Flux du protocole en mode égal à égal

Les schémas ci–après montrent le format d'un échange de communications réussi en mode égal à égal. Le premier illustre un transfert de données de l'équipement source vers l'équipement destinataire (écriture de données) ; le second, un transfert de données de l'équipement destinataire vers l'équipement source (lecture de données).

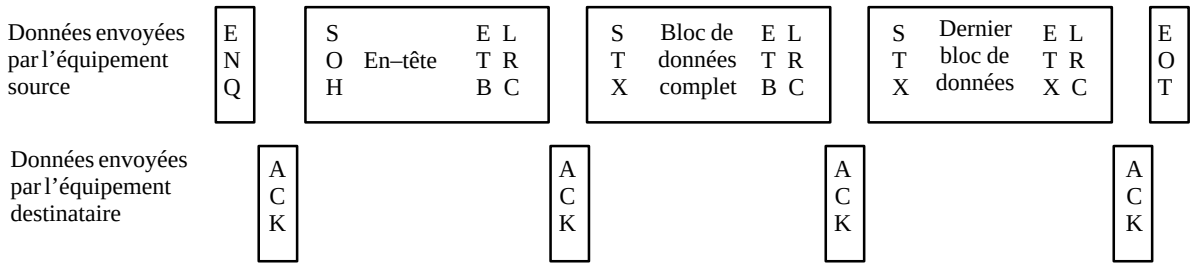


Figure 7-2. Transfert de données de Source vers Destinataire (Ecriture, égal à égal)

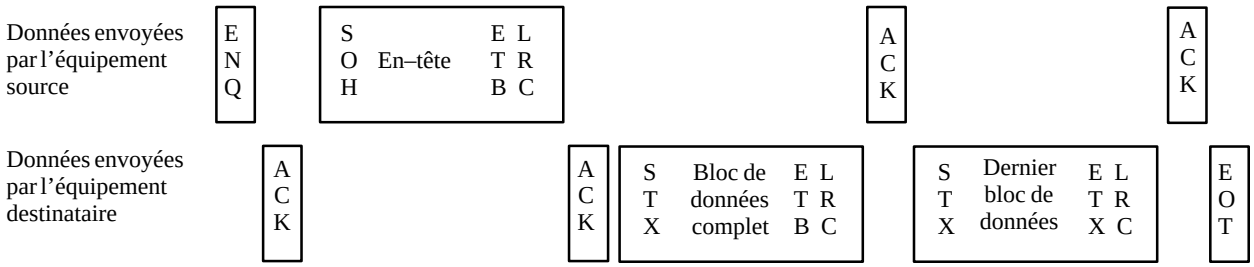


Figure 7-3. Transfert de données de Destinataire vers Source (Lecture, égal à égal)

1.2.1.1. Détection et correction de collisions

En mode d'égal à égal, une collision peut se produire lorsque les deux équipements essaient d'établir la communication en même temps. C'est ce que l'on appelle une collision de demandes. Dans ce cas, chaque équipement attend pendant un certain laps de temps (ENQ_ACK_TIMER) que l'autre lui envoie un ENQ avant d'essayer d'envoyer un autre caractère ENQ. On suppose qu'il y aura collision même si aucun caractère ENQ n'est reçu de l'autre équipement.

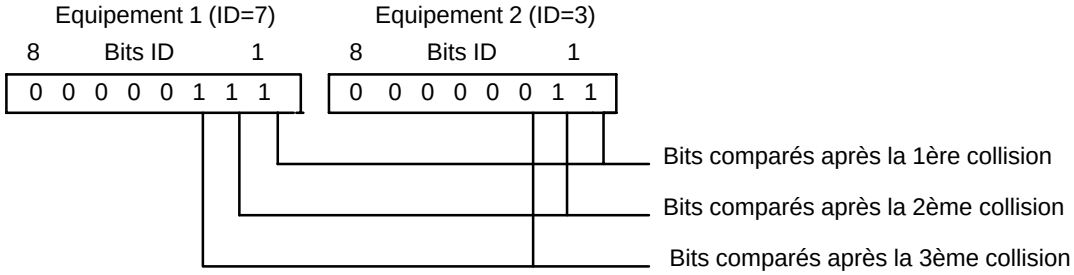
Ce délai est appelé Délai d'attente. Il se base sur l'ID de l'équipement. Chaque ID étant unique, un équipement pourra éventuellement transmettre le caractère ENQ sans qu'il n'y ait de collision.

Le tableau ci–dessous indique les délais d'attente utilisés pour gérer les collisions :

Tableau 7-5. Délais d'attente de ENQ_ACK_TIMER

Vitesse	Bit ID = 0 (Temps en millisecondes)	Bit ID = 1 (Temps en millisecondes)
300	300	440
600	140	220
1200	80	120
2400	80	120
4800	80	120
9600	80	120
19200	80	120

Le schéma ci–après montre la séquence de programmation des délais d'attente qui compare les bits des ID des équipements :


Equipement 1 (ID=7)
8 Bits ID 1
0 0 0 0 0 1 1 1
Equipement 2 (ID=3)
8 Bits ID 1
0 0 0 0 0 0 1 1
Bits comparés après la 1ère collision
Bits comparés après la 2ème collision
Bits comparés après la 3ème collision

1.3. MODE CCM MAÎTRE–ESCLAVE

Ce paragraphe décrit le flux normal du protocole en mode maître–esclave ainsi qu'un échange de séquence Q spécial, uniquement disponible en mode maître–esclave.

1.3.1. Flux normal du protocole

Lors d'une communication maître–esclave normale, la séquence ENQ envoyée par l'équipement maître comporte 3 caractères. Le format et la description de chacun de ces caractères sont fournis ci–dessous :

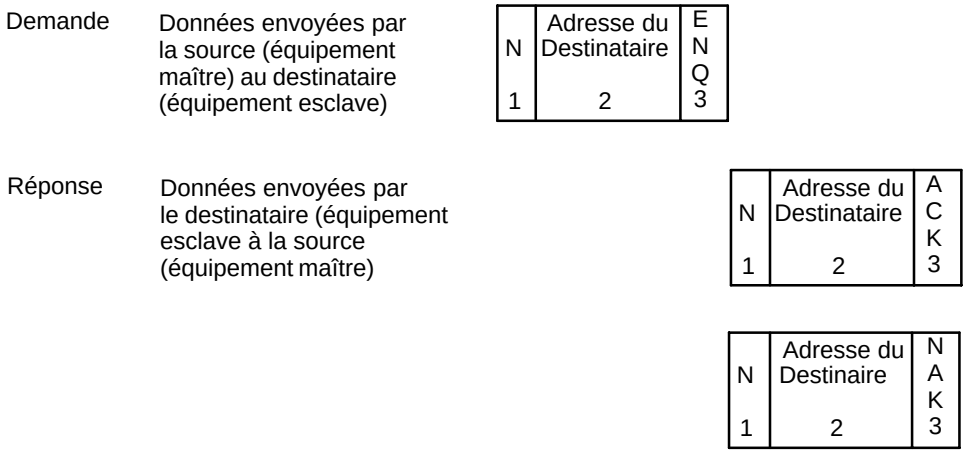


Figure 7-4. Format de la demande ENQ (maître–esclave, Séquence normale)

Les champs de la séquence ENQ sont décrits ci–dessous :

Tableau 7-6. Description de la demande ENQ (maître–esclave, Séquence normale)

Champ	Octet	Description	Gamme valide
N	1	'N' codé ASCII utilisé pour spécifier qu'il s'agit d'une séquence normale.	4Eh
Adresse destinataire	2	ID de l'équipement esclave destinataire + 20h. Exemple : Si l'ID de l'équipement destinataire est 37h. 37h+20h=57h='W'	21h à 7Ah
ENQ	3	3ème caractère de la séquence de demande de 3 caractères.	05h
ACK/NAK	3	Réponse de l'équipement esclave indiquant un accusé de réception positif ou négatif.	06h = ACK 15h = NAK

Le premier schéma ci–dessous montre un transfert de données de l'équipement maître vers l'équipement esclave (écriture de données) ; le second, un transfert de données de l'équipement esclave vers l'équipement maître (lecture de données).

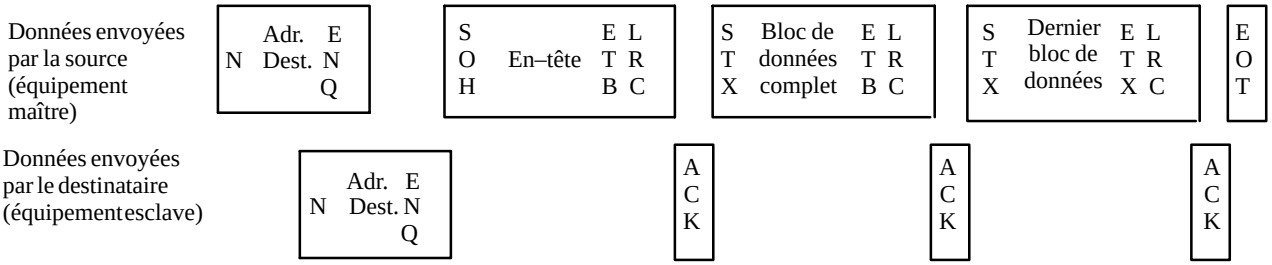


Figure 7-5. Transfert de données du maître vers l'esclave (Ecriture, séquence maître–esclave normale)

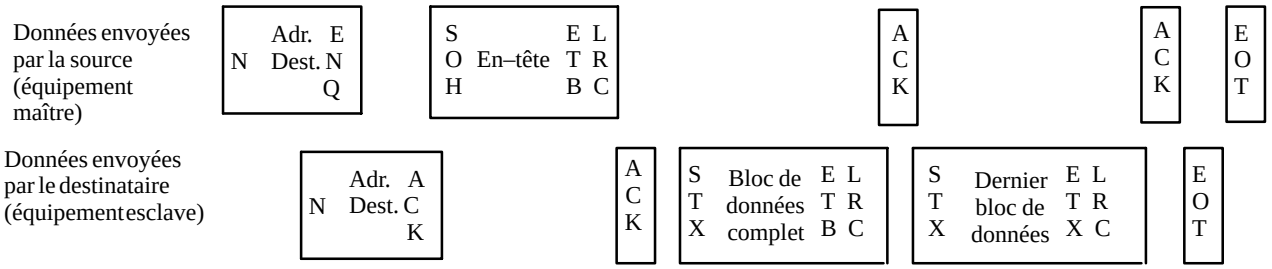


Figure 7-6. Transfert de données de l'esclave vers le maître (Lecture, séquence maître–esclave normale)

1.3.1.1. Echange d'une séquence Q

La séquence Q est un échange spécial du protocole en mode maître–esclave qui permet à l'équipement maître d'extraire 4 octets de données de l'équipement esclave sans générer l'en–tête de 17 octets. Lors d'une communication maître–esclave en Séquence Q, la séquence ENQ de l'équipement maître comporte 3 caractères. La réponse ACK de l'équipement esclave en comporte 8. Le format et la description de l'échange Demande/Données d'une Séquence Q sont illustrés ci–dessous :

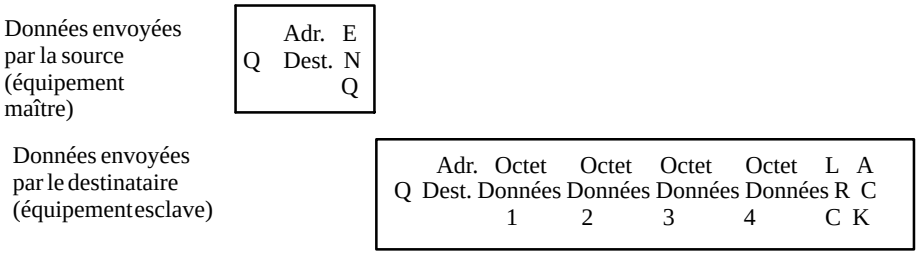

Données envoyées par la source (équipement maître)
Adr. Dest. ENQ
Données envoyées par le destinataire (équipement esclave)
Adr. Dest. Octet Données 1 Octet Données 2 Octet Données 3 Octet Données 4 LRC ACK

Figure 7-7. Transfert de données de l'esclave vers le maître (Séquence Q maître–esclave)

Les tableaux suivants décrivent les différents champs contenus dans les blocs de communication :

Tableau 7-7. Description de la demande du maître (maître–esclave, Séquence Q)

Champ	Octet	Description	Gamme valide
Q	1	'Q' codé ASCII utilisé pour spécifier qu'il s'agit d'une séquence Q.	51h
Adresse destinataire	2	ID de l'équipement esclave destinataire + 20h. Exemple : Si l'ID de l'équipement destinataire est 37h. 37h+20h=57h='W'.	21h à 7Ah
ENQ	3	3ème caractère de la séquence de demande de 3 caractères.	05h

Tableau 7-8. Description de la réponse de l'esclave (maître–esclave, Séquence Q)

Champ	Octet	Description	Gamme valide
Q	1	'Q' codé ASCII utilisé pour spécifier qu'il s'agit d'une séquence Q.	51h
Adresse destinataire	2	ID de l'équipement esclave destinataire + 20h. Exemple : Si l'ID de l'équipement destinataire est 37h. 37h+20h=57h='W'.	21h à 7Ah
Données	3–6	Données dans la réponse à la Séquence Q.	—
LRC	7	Contrôle de redondance longitudinale (LRC). C'est un OU exclusif (XOR) des octets 3–6	—
ACK	8	Caractère de commande ACK	06h

1.4. SPÉCIFICATIONS DE SYNCHRONISATION ET DU NOMBRE D'ESSAIS

Le protocole CCM définit un ensemble d'intervalles de temps et de nombre d'essais à divers niveaux d'échanges. Des valeurs par défaut ont été définies en fonction du protocole. Certains équipements, comme le module CMM Série 90–70, disposent de fonctions permettant de modifier ces valeurs par défaut. Si les valeurs par défaut doivent être modifiées, il faut qu'elles concordent dans les équipements maître et esclave(s).

1.4.1. Temporisations de la liaison série CCM

Les temporisations sont utilisées pour la détection d'erreurs et la reprise sur incident. Chaque fois qu'une temporisation de la liaison série se produit, l'équipement qui la détecte annule les communications en cours et envoie un EOT à l'autre équipement.

Les temporisations définies pour le protocole CCM sont indiquées dans le tableau ci–après. Si un délai d'attente est utilisé, il s'ajoute à la temporisation CCM sur la partie de l'échange concernée. Tous les gestionnaires de communications doivent respecter les temporisations définies ci–après :

Tableau 7-9. Temporisations du protocole CCM

Nom du temporisateur	Description	Sans	Court	Moyen	Long (Défaut)
ENQ_ACK_TIMER	Attendre ACK/NAK du ENQ du maître	0	50	400	800
	Attendre ACK/NAK du ENQ du 1er homologue	0	50	400	800
	Attendre ACK/NAK pour retransmettre ENQ d'un homologue (1)				
	300	0	300/440	300/440	300/440
	600	0	140/220	140/220	140/220
	1200	0	80/120	80/120	80/120
	.				
	.				
	19200	0	80/120	80/120	80/120
SOH_TIMER	Attendre SOH	0	50	400	800
HEADER_TIMER	Attendre Fin d'en–tête				
	300	0	2670	2670	2670
	600	0	1340	1340	1340
	1200	0	670	670	670
	.				
	.				
	19200	0	670	670	670
HEADER_ACK_TIMER	Attendre ACK de l'en–tête	0	50	1000	2000
STX_TIMER	Attendre STX	0	50	10000	20000
DATA_TIMER	Attendre Fin du bloc de données				
	300	0	33340	33340	33340
	600	0	16670	16670	16670
	1200	0	8340	8340	8340
	.				
	.				
	19200	0	8340	8340	8340
DATA_ACK_TIMER	Attendre ACK du bloc de données	0	50	10000	20000
EOT_TIMER	Attendre EOT	0	50	400	800
DELAY_ACK_TIMER	Délai de réponse ACK de l'esclave après ENQ du maître	10 ms + temps pour 4 caractères			
WAIT_FOR_CTS	Attendre CTS	2 secondes			

Le tableau ci–après indique le temps correspondant à la transmission de 4 caractères suivant la vitesse sélectionnée :

Vitesse	Temps de 4 Caractères
300	147 ms
600	74 ms
1200	37 ms
2400	19 ms
4800	10 ms
9600	5 ms
19200	3 ms

1. La valeur dépend de la détection du bit de l'algorithme de délai d'attente dans l'ID CPU. (exemple, à 19200 bps, a un bit 0 attend 80 ms ; un bit 1 attend 120 ms).

1.4.1.1. Nombre d'essais CCM

Le protocole CCM fournit un mécanisme d'essais multiples à divers niveaux d'échanges. Ce mécanisme permet à un équipement de retransmettre des séquences ENQ, des en–têtes ou des blocs de données dont l'accusé de réception a été négatif (NAK).

Le mécanisme d'essais multiples ne doit pas être utilisé si un EOT a été reçu à un quelconque niveau de l'échange. Le tableau ci–dessous indique le nombre d'essais par défaut à chaque niveau d'échange du protocole :

Tableau 7-10. Nombre d'essais définis pour le protocole CCM

Nom du compteur d'essais	Description	Normal (Défaut)	Court
ENQ_COUNT	Comptage des essais de retransmission d'un ENQ d'égal à égal ou maître–esclave	32	3
QSEQ_COUNT	Essais de retransmission d'une séquence Q	3	1
HEADER_COUNT	Comptage des essais de retransmission d'un en–tête	3	1
DATA_BLK COUNT	Essais de retransmission d'un bloc de données	3	1

1.5. TABLES DES ÉTATS DU PROTOCOLE CCM

Les tableaux d'états ci–après illustrent le protocole CCM pour des équipements homologues (mode égal à égal), maîtres et esclaves. Le pseudo–code décrit les actions entreprises dans un état particulier. Des tables d'états et des listes d'actions distinctes sont fournies pour les mises en oeuvre homologues, maîtres et esclaves. Les noms de temporisateurs et de compteurs utilisés dans le pseudo–code correspondent à ceux des temporisateurs et des compteurs mentionnés dans les tableaux Temporisations et Nombre d'essais ci–dessus.

1.5.1. Table des états CCM homologue

Etat/ Evéne-ment	Inactif (1)	Attendre ACK de ENQ (2)	Attendre SOH (3)	Attendre en–tête (4)	Attendre ACK de l'en–tête (5)	Attendre ACK des données (6)	Attendre STX (7)	Attendre données (8)	Attendre EOT (9)	Attendre RP–Q (10)	Collision ENQ (11)
Réception ENQ	Envoi ACK Etat (3) A	Etat (11) B	Envoi EOT Etat (1) K	–	–	–	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K	–	Envoi ACK Etat (3) A
COMM REQ	Envoi ENQ Etat (2) C	–	–	–	–	–	–	–	–	–	–
Réception données API	–	–	–	–	–	Envoi données Etat (6) D	–	–	–	–	–
Réception ACK	–	Envoi en–tête Etat (5) E	Envoi EOT Etat (1) K	–	Etat (6/7) F	Etat (1/6) G	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K	–	Envoi en–tête Etat (5) E
Réception NAK	–	Etat (1/2) H	Envoi EOT Etat (1) K	–	Etat (1/5) I	Etat (1/6) J	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K	–	Etat (1/11) H
Réception EOT	–	Etat (1) L	Etat (1) L	–	Etat (1) L	Etat (1) L	Etat (1) L	–	Etat (1) L	–	Etat (1) L
Réception SOH	–	Envoi EOT Etat (1) K	Etat (4) Q	–	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K
Réception en–tête valide	–	–	–	Envoi ACK Etat(6/7) M	–	–	–	–	–	–	–
Réception STX	–	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Etat (8) R	–	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K
Réception caractère	–	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K
Réception données valides	–	–	–	–	–	–	–	Envoi ACK Etat(7/9) N	–	–	–
En–tête invalides	–	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Etat (1/3) O	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Etat (1/7) P	Envoi EOT Etat (1) K	–	Envoi EOT Etat (1) K
Réception Q–ENQ valide	–	–	–	–	–	–	–	–	–	–	–
Tempo-risation	–	Etat (1/2) H	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	Envoi EOT Etat (1) K	–	Etat (1/2) H

1.5.2. Actions de l'équipement CCM homologue

A Zero all counters
Send ACK to ENQ
Start SOH_TIMER
Transition to State (3)

B Backoff = TRUE
Transition to State (11)

C Zero all counters
Send ENQ
Start ENQ_ACK_TIMER
Transition to State (2)

D Zero DATA_BLK_COUNT
Send Data Block
Start DATA_ACK_TIMER
Remain in State (6)

E Send Header
Start HEADER_ACK_TIMER
Transition to State (5)

F If direction is "read"
Transition to State (7)
Else {
Get data from PLC memory
Send data block
Start DATA_ACK_TIMER
Transition to State (6)
}

G If last block {
Send EOT
Transition to State (1)
}
Else Remain in State (6)

H If ENQ_COUNT <= 32 {
Increment ENQ_COUNT
Send ENQ
Start ENQ_ACK_TIMER
Transition to State (2)
}
Else {
Send EOT
Transition to State (1)
}

I If HEADER_COUNT <= 3 {
Increment HEADER_COUNT
Send Header
Start HEADER_ACK_TIMER
Remain in State (5)
}
Else {
Send EOT
Transition to State (1)
}

J If DATA _BLK_COUNT <= 3 {
Increment DATA_BLK_COUNT
Send data block
Start DATA_ACK_TIMER
Remain in State (6)
}
Else {
Send EOT
Transition to State (1)
}

K Send EOT
Transition to State (1)

L Transition to State (1)

M Send ACK
If "read data" {
Start STX_TIMER
Transition to State (7)
}
Else "write data" {
Get PLC Data
Send Data Block
Start DATA_ACK_TIMER
Transition to State (6)
}

N Send Data to PLC
Send ACK to Data Block
If last_block {
Start EOT_TIMER
Transition to State (9)
}
Else {
Start STX_TIMER
Transition to State (7)
}

O If HEADER_COUNT <= 3 {
Increment HEADER_COUNT
Send NAK
Start SOH_TIMER
Transition to State (3)
}
Else {
Send EOT
Transition to State (1)
}

P If DATA_BLK_COUNT <= 3 {
Increment DATA_BLK_COUNT
Send NAK to bad block
Start STX_TIMER
Transition to State (7)
}
Else {
Send EOT
Transition to State (1)
}

Q Start HEADER_TIMER
Transition to State (4)

R Start DATA_TIMER
Transition to State (8)

1.5.3. Table des états CCM maître

Etat/ Evéne-ment	Inactif (1)	Attendre ACK de ENQ (2)	Attendre SOH (3)	Attendre en–tête (4)	Attendre ACK de l'en–tête (5)	Attendre ACK des données (6)	Attendre STX (7)	Attendre données (8)	Attendre EOT (9)	Attendre RP–Q (10)	Collision ENQ (11)
Réception ENQ	–	–	–	–	–	–	–	–	–	–	–
COMM REQ	Envoi ENQ Etat(2/10) A	–	–	–	–	–	–	–	–	–	–
Réception données API	–	–	–	–	–	Envoi données Etat (6) B	–	–	–	–	–
Réception ACK	–	Envoi en–tête Etat (5) C	–	–	Etat (6/7) D	Etat (1/6) E	Envoi EOT Etat (1) I	–	Envoi EOT Etat (1) I	–	–
Réception NAK	–	Etat (1/2) F	–	–	Etat (1/5) G	Etat (1/6) H	Envoi EOT Etat (1) I	–	Envoi EOT Etat (1) I	–	–
Réception EOT	–	Envoi EOT Etat (1) I	–	–	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	–	Envoi EOT Etat (1) I	–	–
Réception SOH	–	Envoi EOT Etat (1) I	–	–	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	–	Envoi EOT Etat (1) I	–	–
Réception en–tête valide	–	–	–	–	–	–	–	–	–	–	–
Réception STX	–	Envoi EOT Etat (1) I	–	–	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Etat (8) N	–	Envoi EOT Etat (1) I	–	–
Réception caractère	–	Envoi EOT Etat (1) I	–	–	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	–	Envoi EOT Etat (1) I	–	–
Réception données valides	–	–	–	–	–	–	–	Envoi ACK Etat(7/9) K	–	Etat (1) J	–
Réception données/ en–tête invalides	–	Envoi EOT Etat (1) I	–	–	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Etat (1/7) L	Envoi EOT Etat (1) I	Etat (1/10) M	–
Réception Q–ENQ valide	–	–	–	–	–	–	–	–	–	–	–
Tempo-risation	–	Etat (1/2) F	–	–	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Envoi EOT Etat (1) I	Etat (1/10) M	–

1.6. ACTIONS DE L'ÉQUIPEMENT CCM MAÎTRE

A Zero counters
If Q–seq {
Send Q–ENQ–seq
Start ENQ_ACK_TIMER
Transition to State (10)
}
Else {
Send Master–ENQ
Start ENQ_ACK_TIMER
Transition to State (2)
}

B Zero DATA_BLK_COUNT
Send Data Block
Start DATA_ACK_TIMER
Remain in State (6)

C Send Header
Start HEADER_ACK_TIMER
Transition to State (5)

D If direction is "read"
Transition to State (7)
Else "write" {
Get data from PLC memory
Send data block
Start DATA_ACK_TIMER
Transition to State (6)
}

E If last block {
Send EOT
Transition to State (1)
}
Else Remain in State (6)

F If ENQ_COUNT <= 32 {
Increment ENQ_COUNT
Send Master–ENQ
Start ENQ_ACK_TIMER
Remain in State (2)
}
Else {
Send EOT
Transition to State (1)
}

G If HEADER_COUNT <= 3 {
Increment HEADER_COUNT
Send Header
Start HEADER_ACK_TIMER
Remain in State (5)
}
Else {
Send EOT
Transition to State (1)
}

H If DATA_BLK_COUNT <= 3{
Increment DATA_BLK_COUNT
Send data block
Start DATA_ACK_TIMER
Remain in State (6)
}
Else {
Send EOT
Transition to State (1)
}

I Send EOT
Transition to State (1)

J Transition to State (1)

K Send Data to PLC
Send ACK to Data Block
If last_block {
Start EOT_TIMER
Transition to State (9)
}
Else {
Start STX_TIMER
Transition to State (7)
}

L If DATA_BLK_COUNT <= 3{
Increment DATA_BLK_COUNT
Send NAK to bad block
Start STX_TIMER
Transition to State (7)
}
Else {
Send EOT
Transition to State (1)
}

M If QSEQ_COUNT <= 3 {
Increment QSEQ_COUNT
Send Q–ENQ–seq
Start ENQ_ACK_TIMER
Remain in State (10)
}
Else {
Transition to State (1)
}

N Start DATA_TIMER
Transition to State (8)

1.6.1. Table des états CCM esclave

Etat/ Evéne-ment	Inactif (1)	Attendre ACK de ENQ (2)	Attendre SOH (3)	Attendre en–tête (4)	Attendre ACK de l'en–tête (5)	Attendre ACK des données (6)	Attendre STX (7)	Attendre données (8)	Attendre EOT (9)	Attendre RP–Q (10)	Collision ENQ (11)
Réception ENQ	Envoi ACK Etat (3) A	–	Envoi EOT Etat (1) E	–	–	–	Envoi EOT Etat (1) E	–	Envoi EOT Etat (1) E	–	–
COMM REQ	–	–	–	–	–	–	–	–	–	–	–
Réception données API	–	–	–	–	–	Envoi données Etat (6) B	–	–	–	–	–
Réception ACK	–	–	Envoi EOT Etat (1) E	–	–	Etat (1/6) C	Envoi EOT Etat (1) E	–	Envoi EOT Etat (1) E	–	–
Réception NAK	–	–	Envoi EOT Etat (1) E	–	–	Etat (1/6) D	Envoi EOT Etat (1) E	–	Envoi EOT Etat (1) E	–	–
Réception EOT	–	–	Etat (1) F	–	–	Etat (1) F	Etat (1) F	–	Etat (1) F	–	–
Réception SOH	–	–	Etat (4) L	–	–	Envoi EOT Etat (1) E	Envoi EOT Etat (1) E	–	Envoi EOT Etat (1) E	–	–
Réception en–tête valide	–	–	–	Envoi ACK Etat(6/7) G	–	–	–	–	–	–	–
Réception STX	–	–	Envoi EOT Etat (1) E	–	–	Envoi EOT Etat (1) E	Etat (8) M	–	Envoi EOT Etat (1) E	–	–
Réception caractère	–	–	Envoi EOT Etat (1) E	–	–	Envoi EOT Etat (1) E	Envoi EOT Etat (1) E	–	Envoi EOT Etat (1) E	–	–
Réception données valides	–	–	–	–	–	–	–	Envoi ACK Etat(7/9) H	–	–	–
Réception données/ en–tête invalides	–	–	Envoi EOT Etat (1) E	Etat (1/3) I	–	Envoi EOT Etat (1) E	Envoi EOT Etat (1) E	Etat (1/7) J	Envoi EOT Etat (1) E	–	–
Réception Q–ENQ valide	Envoi RP–Q Etat (1) K	–	–	–	–	–	–	–	–	–	–
Tempo-risation	–	–	Envoi EOT Etat (1) E	Envoi EOT Etat (1) E	–	Envoi EOT Etat (1) E	Envoi EOT Etat (1) E	Envoi EOT Etat (1) E	Envoi EOT Etat (1) E	–	–

1.7. ACTIONS DE L'ÉQUIPEMENT CCM ESCLAVE

A DO_ACK=1
Start DELAY_ACK_TIMER
While DELAY_ACK_TIMER != 0{
If char received {
DELAY_ACK_TIMER = 0
DO_ACK = 0
Remain in State (1)
}
}
If (DO_ACK == 1) {
Send 3–char ACK to ENQ
Start SOH_TIMER
Transition to State (3)
}

B Zero DATA_BLK_COUNT
Send Data Block
Start DATA_ACK_TIMER
Remain in State (6)

C If last block {
Send EOT
Transition to State (1)
}
Else wait for PLC data {
Remain in State (6)
}

D If DATA_BLK_COUNT <= 3 {
Increment
DATA_BLK_COUNT
Send Data Block
Start
DATA_ACK_TIMER
Remain in State (6)
}
Else {
Send EOT
Transition to State (1)
}

E Send EOT
Transition to State (1)

F Transition to State (1)

G Send ACK
If "read data" {
Start STX_TIMER
Transition to State (7)
}
Else "write data" {
Get PLC data
Send Data Block
Start
DATA_ACK_TIMER
Transition to State (6)
}

H Send Data to PLC
Send ACK to Data Block
If last–block {
Start EOT_TIMER
Transition to State (9)
}
Else {
Start STX_TIMER
Transition to State (7)
}

I If HEADER_COUNT <= 3 {
Increment
HEADER_COUNT
Send NAK
Start SOH_TIMER
Transition to State (3)
}
Else {
Send EOT
Transition to State (1)
}

J If DATA_BLK_COUNT <= 3 {
Increment
DATA_BLK_COUNT
Send NAK to bad block
Start STX_TIMER
Transition to State (7)
}
Else {
Send EOT
Transition to State (1)
}

K SEND_QRSP = 1
Start T1
While T1 != 0 {
If char received {
T1 = 0
SEND_QRSP = 0
Remain in State (1)
}
}
If (SEND_QRSP ==1) {
Send Response
Remain in State (1)
}

L Start HEADER_TIMER
Transition to State (4)

M Start DATA_TIMER
Transition to State (8)

2. PROTOCOLE RTU

Le protocole de communication série RTU (Remote Terminal Unit) est décrit ci–dessous.

2.1. INTRODUCTION

Lorsque le module de communication série (équipement RTU) est configuré comme un équipement RTU esclave, il utilise le protocole décrit dans ce paragraphe.

Le protocole RTU est un protocole de type interrogation–réponse utilisé pour la communication entre un RTU et un ordinateur–hôte capable de communiquer sous protocole RTU. L'ordinateur–hôte est l'équipement maître ; il envoie une interrogation à un équipement RTU esclave qui lui répond. L'équipement RTU, en tant qu'équipement RTU esclave, ne peut pas interroger ; il ne peut que répondre à l'équipement maître.

Les données RTU transférées consistent en caractères binaires codés sur 8 bits avec un bit de parité en option. Aucun caractère de commande ne figure dans le bloc de données ; un contrôle d'erreurs (contrôle de redondance cyclique CRC) est toutefois inclus comme dernier champ de chaque interrogation et réponse pour garantir la fiabilité de la transmission des données.

2.2. FORMAT DES MESSAGES

Le principe de transfert des messages RTU est illustré ci–dessous :

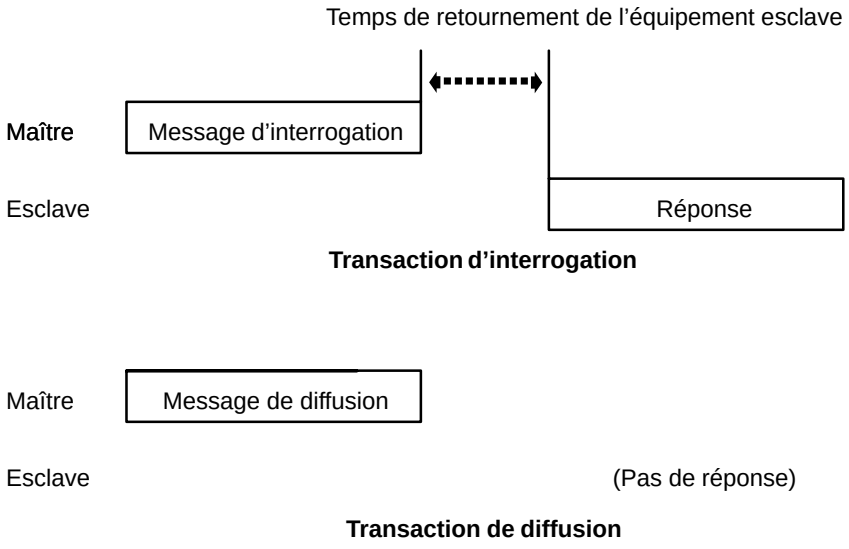


Figure 7-8. Transferts des messages RTU

Une distinction est faite entre les deux équipements qui communiquent. L'équipement qui établit le transfert des données est appelé l'équipement maître et l'autre l'équipement esclave. Le module de communication série (CMM) ne peut être configuré que comme un équipement RTU esclave.

L'équipement maître active le transfert de données en envoyant un message d'interrogation ou de demande de diffusion. L'équipement esclave complète ce transfert de données en envoyant un message de réponse si l'équipement maître lui a adressé un message d'interrogation. Aucun message de réponse n'est retourné si l'équipement maître a envoyé une demande de diffusion. Le temps qui s'écoule entre la fin de l'interrogation et le début de la réponse à cette interrogation est appelé Temps de retournement de l'équipement esclave.

Le temps de retournement illustré ci–dessus dépend de l'interrogation, de l'activité sur l'autre port, et de l'activité du programme d'application de l'API. Une valeur de 500 ms est une estimation raisonnable pour le cas le plus défavorable.

2.3. TYPES DE MESSAGES

Le protocole RTU offre quatre types de messages : interrogation, réponse normale, réponse d'erreur et diffusion.

2.3.1. Interrogation

L'équipement maître envoie un message destiné à un seul équipement esclave.

2.3.1.1. Réponse normale

Une fois que l'équipement esclave a exécuté la fonction spécifiée dans l'interrogation, il renvoie une réponse normale correspondant à cette fonction. Cela signifie que la demande a été satisfaite.

2.3.1.2. Réponse d'erreur

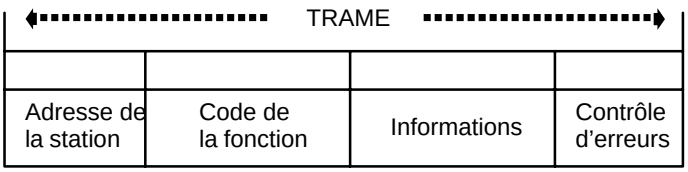
L'équipement esclave reçoit l'interrogation, mais pour une raison ou une autre ne peut pas exécuter la fonction demandée. Dans ce cas, l'équipement esclave renvoie une réponse d'erreur indiquant la raison pour laquelle la demande n'a pas pu être traitée. (Aucun message d'erreur n'est généré pour certains types d'erreurs. Pour plus d'informations à ce sujet, voir paragraphe Erreurs de communication).

2.3.1.3. Diffusion

L'équipement maître envoie un message adressé à tous les équipements esclaves en utilisant l'adresse 0. Tous les équipements esclaves qui reçoivent le message de diffusion exécutent la fonction demandée. Cette transaction prend fin à expiration d'une temporisation définie dans l'équipement maître.

2.4. CHAMPS DES MESSAGES

Les champs contenus dans un message typique sont illustrés ci–dessous :



2.4.1. Adresse de la station

L'adresse de la station est celle de la station esclave sélectionnée pour le transfert de données. Elle occupe un octet et sa valeur est comprise entre 0 et 247 inclus. L'adresse 0 sélectionne toutes les stations esclaves et indique qu'il s'agit d'un message de diffusion. Une adresse entre 1 et 247 sélectionne la station esclave configurée à cette adresse.

2.4.1.1. Code de la fonction

Le code de la fonction identifie la commande adressée à la station. Elle occupe un octet et est définie comme suit pour les valeurs 0 à 255 :

Code de la fonction	Description
0	Fonction illégale
1	Lire table des sorties
2	Lire table des entrées
3	Lire les registres
4	Lire l'entrée analogique
5	Forcer une seule sortie
6	Prédéfinir un seul registre
7	Lire l'état des exceptions
8	Bouclage/Maintenance
9–14	Fonctions non supportées
15	Forcer plusieurs sorties
16	Prédéfinir plusieurs registres
17	Retourner type d'équipement
18–66	Fonctions non supportées
67	Lire mémoire de la zone de travail
68–127	Fonctions non supportées
128–255	Réservées aux réponses négatives

2.4.1.2. Champ Informations

Le champ Informations renferme toutes les informations supplémentaires requises pour mieux spécifier la fonction demandée ou mieux y répondre. Le contenu du champ Informations de chaque type de message (diffusion, interrogation, réponse normale et réponse d'erreur) et tous les codes de fonctions sont fournis dans le paragraphe Description des messages, plus loin dans ce chapitre.

Le champ Informations spécifie les plages de données à accéder dans l'équipement RTU esclave. Le tableau ci–après indique les gammes valides pour tous les modèles d'API GE Fanuc concernés.

Tableau 7-11. Types de mémoire, unités de longueur et gammes valides des équipements esclaves

Type de mémoire de l'API	Unité de longueur	Gamme valide (1)
Registres (%R)	mot	0 à maximum – 1
Entrées analogiques (%AI)	mot	0 à maximum – 1
Entrées logiques (%I)	bit	0 à maximum – 1
	octet	0 à maximum – 1
Sorties logiques (%Q)	bit	0 à maximum – 1
	octet	0 à maximum – 1

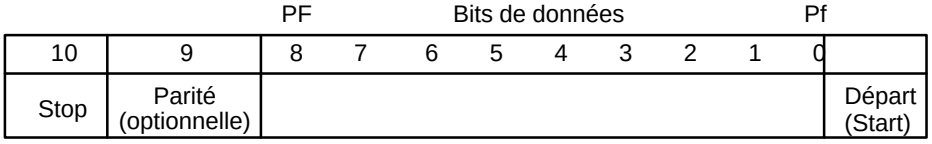
1 Les gammes maximales adressables pour chaque type de mémoire dépendent du modèle de CPU et de la configuration de la mémoire.

2.4.1.3. Champ Contrôle d'erreurs

Le champ Contrôle d'erreurs occupe deux octets et renferme un code de redondance cyclique (CRC–16). Sa valeur varie suivant le contenu de l'adresse de la station, du code de la fonction et du champ Informations. La façon dont le code CRC–16 est généré est expliquée dans le paragraphe Contrôle de redondance cyclique (CRC), plus loin dans ce chapitre. La longueur du champ Informations étant variable, la longueur de la trame doit être déterminée pour que le code CRC–16 soit généré correctement. Pour plus d'informations à ce sujet, voir paragraphe Calcul de la longueur de la trame, plus loin dans ce chapitre.

2.5. FORMAT DES CARACTÈRES

Les messages sont envoyés comme une suite de caractères. Chaque octet dans le message est transmis comme un caractère. Le schéma ci–dessous montre le format des caractères. Un caractère comprend un bit de Start (0), huit bits de données, un bit de parité optionnel et un bit de stop (1). Entre les caractères, la ligne est maintenue à l'état 1.



2.6. FIN DU MESSAGE

Chaque station surveille le temps qui s'écoule entre deux caractères. Lorsque le temps correspondant à la transmission de trois caractères s'est écoulé sans qu'aucun nouveau caractère n'ait été reçu, elle considère que le message est terminé. Le prochain caractère reçu sera interprété comme le début d'un nouveau message.

La fin d'une trame coïncide avec le premier des deux événements suivants :

- Le nombre de caractères reçus pour la trame correspond à la longueur de trame calculée.
- Le temps correspondant à la transmission de 3 caractères s'est écoulé sans qu'aucun nouveau caractère n'ait été reçu.

2.7. UTILISATION DES TEMPORISATIONS

Les temporisations sont utilisées sur la liaison série pour détecter les erreurs, faire une reprise sur incident et éviter que la fin des messages et des séquences de messages ne soit tronquée. Bien que le module autorise un délai

correspondant à la transmission de trois caractères entre chaque caractère du message qu'il reçoit, il transmet les caractères du message qu'il envoie avec un délai de moins d'un demi–caractère entre chaque.

Après avoir envoyé un message d'interrogation, l'équipement maître doit attendre environ 500 millisecondes avant de considérer que l'équipement esclave n'a pas répondu à sa demande.

2.8. CONTRÔLE DE REDONDANCE CYCLIQUE (CRC)

Le Contrôle de Redondance Cyclique (CRC) est l'une des méthodes de contrôle d'erreurs les plus efficaces. Le CRC consiste en 2 caractères de contrôle générés par l'émetteur et ajoutés à la fin des caractères de données transmis. Utilisant la même méthode, le récepteur génère son propre CRC sur les données entrantes et le compare à celui transmis par l'émetteur pour s'assurer de la fiabilité du transfert.

Le calcul mathématique complet du CRC n'est pas expliqué ici. Vous trouverez cette information dans de nombreux ouvrages consacrés à la communication des données. Toutefois, les points essentiels à retenir dans le calcul du CRC sont les suivants :

- Les bits de données qui constituent le message sont multipliés par le nombre de bits dans le CRC.
- Le résultat obtenu est alors divisé par le polynôme générateur (utilisant un modulo 2 sans retenues). Le CRC correspond au reste de cette division.
- Le quotient est ignoré et le reste (CRC) est ajouté aux bits des données, puis le message est transmis avec le CRC.
- Le récepteur divise alors le message plus le CRC par le polynôme générateur et si le reste est nul, alors la transmission s'est effectuée sans erreur.

Un polynôme générateur est une expression algébrique fournie sous forme d'une chaîne de termes X élevés à des puissances telle que $X^3 + X^2 + X^0$ (ou 1) qui peut être exprimée sous la forme binaire 1101. Un polynôme générateur peut avoir n'importe quelle longueur et contenir n'importe quelle suite de 0 et de 1, du moment que l'émetteur et le récepteur utilisent la même valeur. Pour une détection d'erreurs optimale, certains polynômes générateurs standard ont toutefois été développés. Le protocole RTU utilise le polynôme $X^{16} + X^{15} + X^2 + 1$ qui en binaire donne 1 1000 0000 0000 0101. Le CRC que génère ce polynôme est connu sous le nom de CRC–16.

Le mécanisme expliqué ci–dessus peut être mis en oeuvre au niveau matériel ou logiciel. Une des mises en oeuvre matérielles consiste à fabriquer un registre à décalage à plusieurs segments basé sur le polynôme générateur.

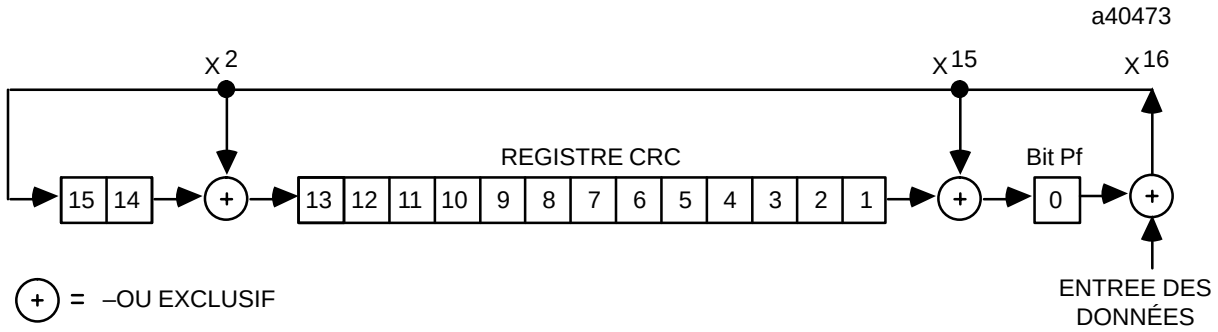


Figure 7-9. Registre de Contrôle de Redondance Cyclique (CRC)

Pour générer le CRC, les bits de données du message sont entrés un par un dans le registre à décalage. Le registre CRC renferme une valeur prédéfinie. Au fur et à mesure que les bits de données sont entrés dans le registre, les bits déjà présents sont décalés vers la droite. Le bit de poids faible (Pf) est combiné par une porte OU exclusif (XOR) avec le bit de données et le résultat est combiné par une porte OU exclusif (XOR) avec l'ancien contenu du bit 1 (le résultat est placé dans le 0), puis avec l'ancien contenu du bit 14 (le résultat est placé dans le bit 13), et pour finir, il est décalé dans le bit 15. Ce processus est répété jusqu'à ce que tous les bits de données du message aient été traités. La mise en oeuvre logicielle du CRC–16 est décrite dans le paragraphe qui suit.

2.8.1. Calcul du CRC–16

Le pseudo–code pour le calcul du CRC–16 est fourni ci–dessous :

```
                    Prédéfinir le nombre d'octets en fonction des données à transmettre.
                    Initialiser les 16 bits du registre du reste (CRC) à 1.
                    Combiner par une porte OU exclusif (XOR) les 8 bits du premier octet de données avec
                        l'octet de poids fort du registre CRC à 16 bits. Le résultat est le CRC courant.
DECALAGE INITIAL :  Initialiser le compteur de décalages à 0.
DECALAGE :          Décaler le registre CRC courant de 1 bit vers la droite.
                    Incrémenter le compteur de décalages.
                    Le bit éliminé par le décalage à droite (indicateur) est–il un 1 ou un 0 ?
                        Si c'est un 1, combiner par une porte OU exclusif (XOR) le polynôme générateur
                            avec le CRC courant.
                        Si c'est un 0, continuer.
                    Le compteur de décalages indique–t–il 8 ?
                        Si NON, revenir à DECALAGE.
                        Si OUI, incrémenter le nombre d'octets.
                    Le nombre d'octets est–il supérieur à la longueur des données ?
                        Si NON, combiner par une porte OU exclusif (XOR) les 8 bits de l'octet de
                            données suivant avec le CRC courant et aller à DECALAGE INITIAL.
                        Si OUI, ajouter le CRC courant à la fin du message de données à transmettre et
                            sortir.
```

Une fois le message transmis, le récepteur effectuera la même opération CRC sur tous les bits de données et le CRC transmis. Si les données ont été transmises correctement, le reste obtenu (CRC du récepteur) doit être égal à 0.

2.8.2. Exemple de calcul CRC–16

L'équipement RTU envoie d'abord l'octet de poids faible (des registres ou des données logiques). Le premier bit du CRC–16 transmis est le bit de poids fort. C'est pourquoi, dans l'exemple fourni, le bit de poids fort du polynôme CRC est à l'extrême–droite. Le terme X^{16} est abandonné car il n'affecte que le quotient (qui est ignoré) et pas le reste (caractères CRC). Le polynôme générateur est par conséquent 1010 0000 0000 0001. Tous les bits du reste sont initialisés à 1.

A titre d'exemple, nous calculerons le CRC–16 du message RTU Lire état des exceptions (07). Le format du message est le suivant :

Adresse	Fonction	CRC–16
01	07	

Dans cet exemple, c'est l'équipement numéro 1 qui est interrogé (adresse 01). Pour calculer le CRC, le volume des données à transmettre doit être connu ; cette information est fournie par chaque type de message dans le paragraphe Calcul de la longueur de la trame, plus loin dans ce chapitre. Pour ce message, les données occupent 2 octets.

ALGORITHME CRC–16 DE L'EMETTEUR

	PF[2]	Pf[2] Indicateur
Reste initial	1111 1111 1111 1111	
XOR 1er octet de données	0000 0000 0000 0001	
CRC courant	1111 1111 1111 1110	
Décalage 1	0111 1111 1111 1111	0
Décalage 2	0011 1111 1111 1111	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1001 1111 1111 1110	
Décalage 3	0100 1111 1111 1111	0
Décalage 4	0010 0111 1111 1111	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1000 0111 1111 1110	
Décalage 5	0100 0011 1111 1111	0
Décalage 6	0010 0001 1111 1111	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1000 0001 1111 1110	
Décalage 7	0100 0000 1111 1111	0
Décalage 8	0010 0000 0111 1111	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1000 0000 0111 1110	
XOR 2ème octet de données	0000 0000 0000 0111	
CRC courant	1000 0000 0111 1001	
Décalage 1	0100 0000 0011 1100	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1110 0000 0011 1101	
Décalage 2	0111 0000 0001 1110	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1101 0000 0001 1111	
Décalage 3	0110 1000 0000 1111	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1100 1000 0000 1110	
Décalage 4	0110 0100 0000 0111	0
Décalage 5	0011 0010 0000 0011	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1001 0010 0000 0010	
Décalage 6	0100 1001 0000 0001	0
Décalage 7	0010 0100 1000 0000	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC courant	1000 0100 1000 0001	
Décalage 8	0100 0010 0100 0000	1
XOR polynôme gén.	1010 0000 0000 0001	
CRC transmis	1110 0010 0100 0001	
	E 2 4 1	

ALGORITHME CRC–16 DU RECEPTEUR[1]

	PF[2]	Pf[2] Indicateur
CRC Rcpt après données	1110 0010 0100 0001	
XOR 1er octet CRC trns	0000 0000 0100 0001	
CRC courant	1110 0010 0000 0000	
Décalage 1	0111 0001 0000 0000	0
Décalage 2	0011 1000 1000 0000	0
Décalage 3	0001 1100 0100 0000	0
Décalage 4	0000 1110 0010 0000	0
Décalage 5	0000 0111 0001 0000	0
Décalage 6	0000 0011 1000 1000	0
Décalage 7	0000 0001 1100 0100	0
Décalage 8	0000 0000 1110 0010	0
XOR 2ème octet CRC trns	0000 0000 1110 0010	
CRC courant	0000 0000 0000 0000	
Décalage résultats 1–8	0000 0000 0000 0000	

TOUS LES BITS A ZERO DANS LE CRC–16 FINAL DU RECEPTEUR INDIQUENT UNE TRANSMISSION CORRECTE !

EXEMPLE DE MESSAGE
Voir exemple de message transmis ci–dessous.

[1] Comme indiqué précédemment, le récepteur traite les données en entrée en utilisant le même algorithme CRC que l'émetteur. L'exemple donné pour le récepteur commence lorsque tous les bits de données, exception faite du CRC transmis, ont été reçus correctement.C'est la raison pour laquelle, à ce stade, le CRC du récepteur est égal à celui qui a été transmis. Dans ce cas, le résultat de l'algorithme CRC est zéro, ce qui signifie que la transmission est correcte.
Le message transmis avec le CRC devrait alors être le suivant :

Adresse	Fonction	CRC–16	
01	07	41	E2

[2] Les bits de poids faible et fort mentionnés ne s'appliquent qu'aux octets de données, et *non pas* aux octets CRC. L'ordre des bits de poids faible et fort du CRC est l'inverse de celui des octets de données.

2.9. CALCUL DE LA LONGUEUR DE LA TRAME

Pour générer le CRC–16 d'un message, la longueur du message doit être connue. La longueur de tous les types de messages peut être déterminée à partir du tableau ci–dessous.

Tableau 7-12. Longueur des messages RTU

Code et nom de la fonction		Longueur du message d'interrogation ou de diffusion moins code CRC	Longueur du message de réponse moins code CRC
0		Non définie	Non définie
1	Lire Table des sorties	6	3 + 3ème octet (1)
2	Lire Table des entrées	6	3 + 3ème octet (1)
3	Lire les registres	6	3 + 3ème octet (1)
4	Lire Entrée analogique	6	3 + 3ème octet (1)
5	Forcer une seule sortie	6	6
6	Prédéfinir un seul registre	6	6
7	Lire état des exceptions	2	3
8	Bouclage/Maintenance	6	6
9–14		Non définies	Non définies
15	Forcer plusieurs sorties	7 + 7ème octet (1)	6
16	Prédéfinir plusieurs registres	7 + 7ème octet (1)	6
17	Retourner Type d'équipement	2	8
18–66		Non définies	Non définies
67	Lire Zone de travail	6	3 + 3ème octet (1)
68–127		Non définies	Non définies
128–255		Non définies	3

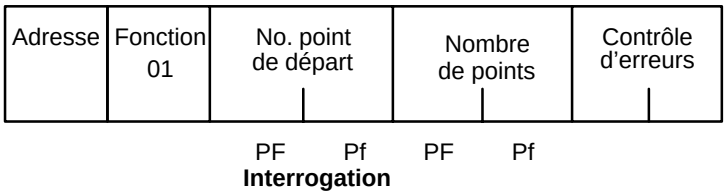
[1] La valeur de cet octet correspond au nombre d'octets contenus dans les données transmises.

2.10. DESCRIPTION DES MESSAGES

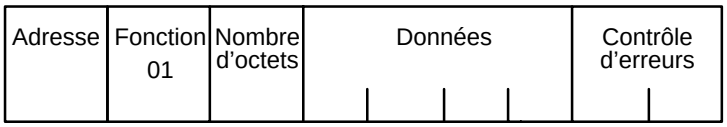
Vous trouverez dans les pages qui suivent la description du format et des champs de chaque message RTU.

MESSAGE (01) : **LIRE TABLE DES SORTIES**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

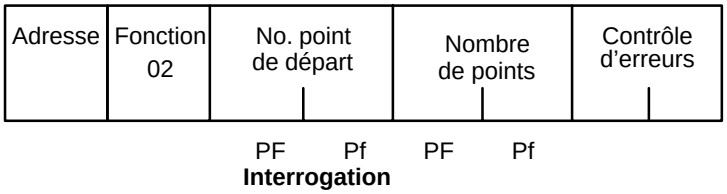
- L'**adresse** 0 n'est pas autorisée dans la mesure où ce message ne peut pas être une demande de diffusion.
- Le **code de la fonction** est 01.
- Le **numéro du point de départ** occupe deux octets et peut prendre n'importe quelle valeur, du moment qu'elle est inférieure au numéro du point de sortie le plus haut disponible dans la CPU Série 90 connectée. Le numéro du point de départ est égal au numéro du premier point de sortie retourné dans la réponse normale moins 1.
- La valeur **nombre de points** occupe deux octets. Elle spécifie le nombre de points retournés dans la réponse normale. La somme des valeurs Numéro du point de départ et Nombre de points doit être inférieure ou égale au numéro du point de sortie le plus haut disponible dans la CPU Série 90 connectée. L'octet de poids fort des champs Numéro du point de départ et Nombre de points est transmis en premier, suivi de l'octet de poids faible de ces deux champs.

REPONSE :

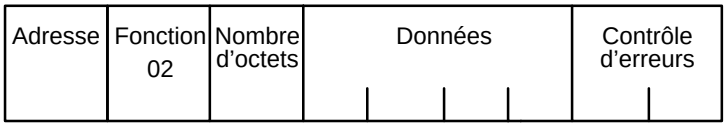
- Le **nombre d'octets** est un nombre binaire compris entre 0 et 255 (0=256). C'est le nombre d'octets figurant dans la réponse normale entre le champ Nombre d'octets et le champ Contrôle d'erreurs.
- Le **champ données** de la réponse normale contient l'état des sorties au format condensé. Chaque octet renferme 8 valeurs de points de sortie. Le bit de poids faible (Pf) du premier octet renferme la valeur du point de sortie correspondant au numéro du point de départ plus 1. Les valeurs des points de sortie sont classées par numéro d'ordre, en commençant par le bit de poids faible du premier octet du champ de données et en finissant par le bit de poids fort (PF) du dernier octet du champ de données. Si le nombre de points n'est pas un multiple de 8, le dernier octet de données renferme des zéros dans un à sept des bits de poids fort.

MESSAGE (02) : **LIRE TABLE DES ENTREES**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

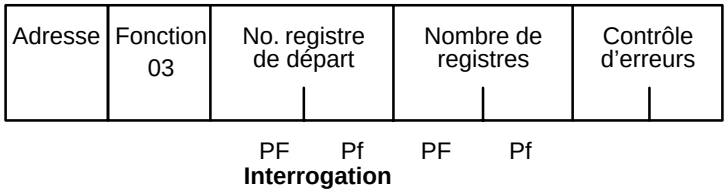
- L'**adresse** 0 n'est pas autorisée dans la mesure où ce message ne peut pas être une demande de diffusion.
- Le **code de la fonction** est 02.
- Le **numéro du point de départ** occupe deux octets et peut prendre n'importe quelle valeur, du moment qu'elle est inférieure au numéro du point d'entrée le plus haut disponible dans la CPU Série 90 connectée. Le numéro du point de départ est égal au numéro du premier point d'entrée retourné dans la réponse normale moins 1.
- La valeur **nombre de points** occupe deux octets. Elle spécifie le nombre de points retourné dans la réponse normale. La somme des valeurs Numéro du point de départ et Nombre de points doit être inférieure ou égale au numéro du point de sortie le plus haut disponible dans la CPU Série 90 connectée. L'octet de poids fort des champs Numéro du point de départ et Nombre de points est transmis en premier, suivi de l'octet de poids faible de ces deux champs.

REPONSE :

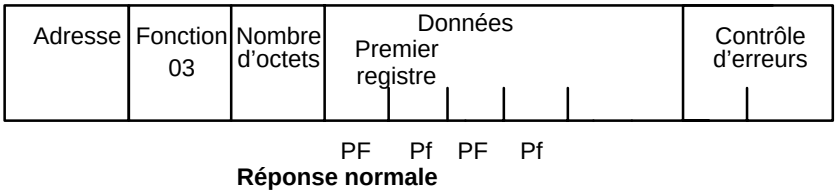
- Le **nombre d'octets** est un nombre binaire compris entre 0 et 255 (0=256). C'est le nombre d'octets figurant dans la réponse normale entre le champ Nombre d'octets et le champ Contrôle d'erreurs.
- Le **champ données** de la réponse normale contient l'état des entrées au format condensé. Chaque octet renferme 8 valeurs de points d'entrée. Le bit de poids faible (Pf) du premier octet renferme la valeur du point d'entrée correspondant au numéro du point de départ plus 1. Les valeurs des points d'entrée sont classées par numéro d'ordre, en commençant par le bit de poids faible du premier octet du champ de données et en finissant par le bit de poids fort (PF) du dernier octet du champ de données. Si le nombre de points n'est pas un multiple de 8, le dernier octet de données renferme des zéros dans un à sept des bits de poids fort.

MESSAGE (03) : **LIRE LES REGISTRES**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

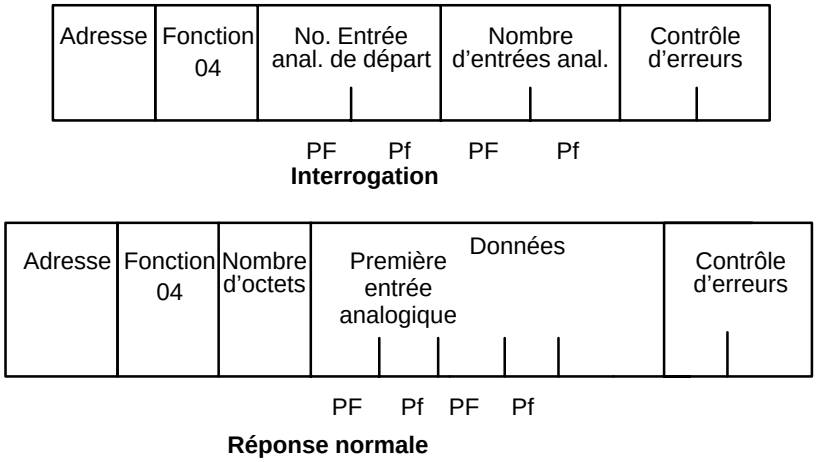
- L'**adresse** 0 n'est pas autorisée dans la mesure où ce message ne peut pas être une demande de diffusion.
- Le **code de la fonction** est 03.
- Le **numéro du registre de départ** occupe deux octets et peut prendre n'importe quelle valeur, du moment qu'elle est inférieure au numéro du registre le plus haut disponible dans la CPU Série 90 connectée. Le numéro du registre de départ est égal au numéro du premier registre retourné dans la réponse normale moins 1.
- La valeur **nombre de registres** occupe deux octets. Elle doit être comprise entre 1 et 125 inclus. La somme des valeurs Numéro du registre de départ et Nombre de registres doit être inférieure ou égale au numéro du registre le plus haut disponible dans la CPU Série 90 connectée. L'octet de poids fort des champs Numéro du registre de départ et Nombre de registres est transmis en premier, suivi de l'octet de poids faible de ces deux champs.

REPONSE :

- Le **nombre d'octets** est un nombre binaire compris entre 2 et 250 inclus. C'est le nombre d'octets figurant dans la réponse normale entre le champ Nombre d'octets et le champ Contrôle d'erreurs. Le nombre d'octets est égal à deux fois le nombre de registres retourné dans la réponse. Le nombre d'octets est limité à 250 (125 registres) de façon que la réponse puisse tenir dans un seul bloc de données de 256 octets.
- Les registres sont classés par numéro d'ordre dans le champ des données ; le registre ayant le plus petit numéro dans les deux premiers octets et celui ayant le plus grand numéro dans les deux derniers octets du champ de données. Le numéro du premier registre dans le champ de données correspond au numéro du registre de départ plus 1. L'octet de poids fort de chaque registre est transmis avant l'octet de poids faible.

MESSAGE (04) : **LIRE LES ENTREES ANALOGIQUES**

FORMAT :



INTERROGATION :

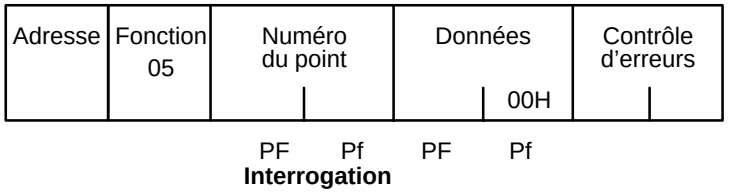
- L'**adresse** 0 n'est pas autorisée dans la mesure où ce message ne peut pas être une demande de diffusion.
- Le **code de la fonction** est 04.
- Le numéro de l'**entrée analogique de départ** occupe deux octets et peut prendre n'importe quelle valeur, du moment qu'elle est inférieure au numéro de l'entrée analogique le plus haut disponible dans la CPU Série 90 connectée. Le numéro de l'entrée analogique de départ est égal au numéro de la première entrée analogique retourné dans la réponse normale moins 1.
- La valeur **nombre d'entrées analogiques** occupe deux octets. Elle doit être comprise entre 1 et 125 inclus. La somme des valeurs Numéro de l'entrée analogique de départ et Nombre d'entrées analogiques doit être inférieure ou égale au numéro de l'entrée analogique le plus haut disponible dans la CPU Série 90 connectée. L'octet de poids fort des champs Numéro de l'entrée analogiques de départ et Nombre d'entrées analogiques est transmis en premier, suivi de l'octet de poids faible de ces deux champs.

REPONSE :

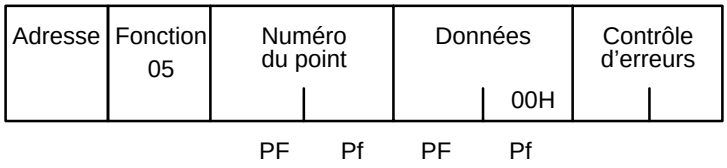
- Le **nombre d'octets** est un nombre binaire compris entre 2 et 250 inclus. C'est le nombre d'octets figurant dans la réponse normale entre le champ Nombre d'octets et le champ Contrôle d'erreurs. Le nombre d'octets est égal à deux fois le nombre d'entrées analogiques retourné dans la réponse. Le nombre d'octets est limité à 250 (125 entrées analogiques) de façon que la réponse puisse tenir dans un seul bloc de données de 256 octets.
- Les entrées analogiques sont classées par numéro d'ordre dans le champ de données ; l'entrée analogique ayant le plus petit numéro dans les deux premiers octets et celle ayant le plus grand numéro dans les deux derniers octets du champ de données. Le numéro de la première entrée analogique dans le champ de données correspond au numéro d'entrée analogique de départ plus 1. L'octet de poids fort de chaque entrée analogique est transmis avant l'octet de poids faible.

MESSAGE (05) : **FORCER UNE SEULE SORTIE**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

- L'**adresse** 0 indique une demande de diffusion. Toutes les stations esclaves traitent les demandes de diffusion mais sans renvoyer de réponse.
- Le **code de la fonction** est 05.
- Le champ **numéro du point** occupe deux octets. Il est égal au numéro du point de sortie à forcer moins 1.
- Le premier octet du champ de **données** indique 0 ou 255 (FFH). Le point de sortie spécifié dans le champ numéro du point sera forcé à 0 si le premier octet du champ de données est égal à 0 ; il sera forcé à 1 si le premier octet du champ de données est égal à 255 (FFH). Le second octet du champ de données est toujours égal à 0.

REPONSE :

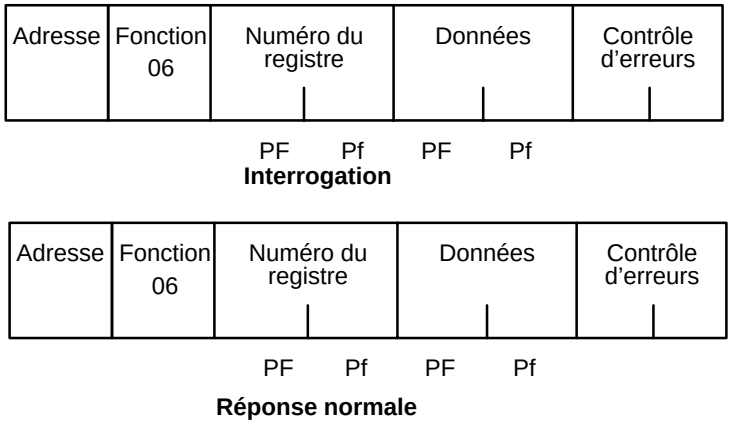
- La réponse normale à une demande Forcer une seule sortie est identique à l'interrogation.

Remarque

La demande Forcer une seule sortie n'est pas une commande de forçage permanente. La sortie spécifiée dans cette demande ne sera forcée à la valeur indiquée qu'au début d'un seul cycle de l'application Série 90.

MESSAGE (06) : **PREDEFINIR UN SEUL REGISTRE**

FORMAT :



INTERROGATION :

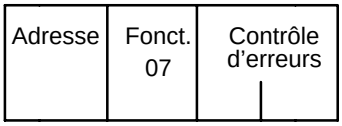
- L'**adresse** 0 indique une demande de diffusion. Toutes les stations esclaves traitent les demandes de diffusion mais sans renvoyer de réponse.
- Le **code de la fonction** est 06.
- Le champ **numéro du registre** occupe deux octets et peut prendre n'importe quelle valeur, du moment qu'elle est inférieure au numéro du registre le plus haut disponible dans la CPU Série 90 connectée. Il est égal au numéro du registre à prédéfinir à forcer moins 1.
- Le champ **données** occupe deux octets et renferme la valeur avec laquelle le registre spécifié dans le champ Numéro du registre doit être prédéfini. Le premier octet du champ de données renferme l'octet de poids fort de la valeur prédéfinie. Le second octet du champ de données renferme l'octet de poids faible.

REPONSE :

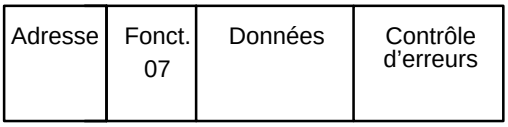
- La réponse normale à une demande Prédéfinir un seul registre est identique à l'interrogation.

MESSAGE (07) : **LIRE ETAT DES EXCEPTIONS**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

Cette interrogation est une forme abrégée permettant de lire les huit premiers points de sortie.

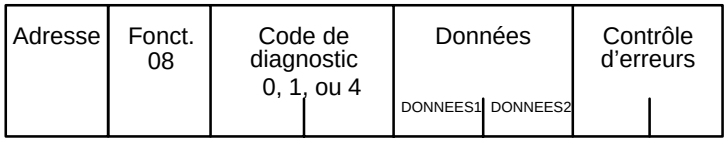
- L'**adresse** 0 n'est pas autorisée dans la mesure où ce message ne peut pas être une demande de diffusion.
- Le **code de la fonction** est 07.

REPONSE :

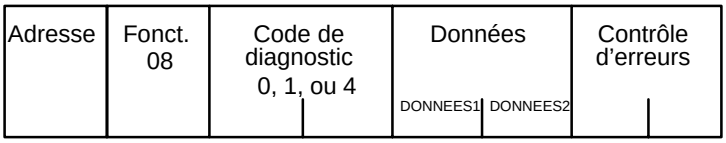
- Le champ **données** de la réponse normale occupe un seul octet et renferme l'état des points de sortie un à huit. Les états des sorties sont classés par numéro d'ordre ; état du premier point de sortie dans le bit de poids faible et état du point de sortie huit dans le bit de poids fort.

MESSAGE (08) : **BOUCLAGE/MAINTENANCE (CARACTERE GENERAL)**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

- Le **code de la fonction** est 08.
- Le **code de diagnostic** occupe deux octets. L'octet de poids fort du code de diagnostic est le premier octet transmis dans le champ Code de diagnostic. L'octet de poids faible est le deuxième octet transmis. La commande Bouclage/Maintenance n'est définie que pour les codes de diagnostic 0, 1 ou 4. Tous les autres codes de diagnostic sont réservés.
- Le champs **données** occupe deux octets. Le contenu des deux octets de données est déterminé par la valeur du code de diagnostic.

REPONSE :

- Voir description des différents codes de diagnostic ci–après :

CODE DE DIAGNOSTIC (00) **Retourner données de l'interrogation (Bouclage/Maintenance)** :

- Une interrogation de bouclage/maintenance ayant le code de diagnostic 0 est une demande de retour des données de l'interrogation.
- L'**adresse** 0 n'est pas autorisée pour cette demande.
- Les valeurs des deux octets du champ **données** sont arbitraires.
- La réponse normale est identique à l'interrogation.
- Les valeurs des octets de **données** dans la réponse sont identiques à celles envoyées dans l'interrogation.

CODE DE DIAGNOSTIC (01) **Provoquer un redémarrage des communications (Bouclage/Maintenance)**

Une demande de bouclage/maintenance (de type interrogation ou diffusion) ayant le code de diagnostic 1 est une demande Provoquer un redémarrage des communications.

- L'**adresse** 0 indique une demande de diffusion. Toutes les stations esclaves traitent les demandes de diffusion mais sans renvoyer de réponse.
- Cette demande désactive le mode Réception seulement (permet aux stations esclaves d'envoyer des réponses lorsque des interrogations sont reçues pour pouvoir redémarrer les communications).
- La valeur du premier octet du champ **données** (DONNEES1) doit être 0 ou FF. Toute autre valeur génèrera une réponse d'erreur. La valeur du second octet du champ de données (DONNEES2) est toujours égale à 0.
- La réponse normale à une demande Provoquer le redémarrage des communications est identique à l'interrogation.

CODE DE DIAGNOSTIC (04) **Forcer mode Réception seulement (Bouclage/Maintenance)**

Une demande de bouclage/maintenance (de type interrogation ou diffusion) ayant le code de diagnostic 4 est une demande de forçage du mode Réception seulement.

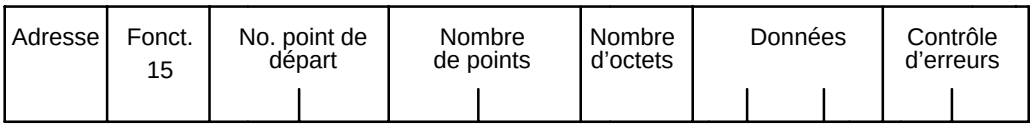
- L'**adresse** 0 indique une demande de diffusion. Toutes les stations esclaves traitent les demandes de diffusion.
- A réception d'une demande Forcer mode Réception seulement, l'équipement RTU se configurera en mode Réception seulement, n'exécutera pas les fonctions demandées et n'enverra aucune réponse (ni normale ni d'erreur) aux interrogations. Le mode Réception seulement est désactivé lorsque l'équipement RTU reçoit une demande Provoquer le redémarrage des communications ou lorsque l'équipement est redémarré.
- Les deux octets du champ **données** d'une demande Forcer mode Réception seulement sont à 0. L'équipement RTU ne répond jamais à une demande Forcer mode Réception seulement.

Remarque

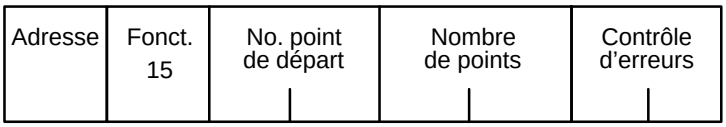
Lorsqu'il est redémarré, l'équipement RTU désactive le mode Réception seulement et peut de nouveau répondre aux interrogations.

MESSAGE (15) : **FORCER PLUSIEURS SORTIES**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

- L'**adresse** 0 indique une demande de diffusion. Toutes les stations esclaves traitent les demandes de diffusion mais sans renvoyer de réponse.
- Le **code de la fonction** est 15.
- Le champ **numéro du point de départ** occupe deux octets et peut prendre n'importe quelle valeur, du moment qu'elle est inférieure au numéro du point de sortie le plus haut disponible dans la CPU Série 90 connectée. Le numéro du point de départ est égal l est égal au numéro du premier point de sortie forcé par cette demande.
- La valeur **nombre de points** occupe deux octets. La somme des valeurs Numéro du point de départ et Nombre de points doit être inférieure ou égale au numéro du point de sortie le plus haut disponible dans la CPU Série 90 connectée. L'octet de poids fort des champs Numéro du point de départ et Nombre d'octets est transmis en premier, suivi de l'octet de poids faible de ces deux champs.
- Le **nombre d'octets** est un nombre binaire compris entre 0 et 255 (0=256). C'est le nombre d'octets figurant dans le champ de données de la demande Forcer plusieurs sorties.
- Le champ **données** renferme les valeurs auxquelles les sorties spécifiées dans les champs Numéro du point de départ et Nombre de points doivent être forcées. Chaque octet du champ de données renferme les valeurs auxquelles huit points de sortie doivent être forcés. Le bit de poids faible (Pf) du premier octet renferme la valeur à laquelle le point de sortie dont le numéro est égal à celui du point de départ plus 1 doit être forcé. Les valeurs des points de sortie sont classées par numéro d'ordre, en commençant par le bit de poids faible du premier octet du champ de données et en finissant par le bit de poids fort (PF) du dernier octet du champ de données. Si le nombre de points n'est pas un multiple de 8, le dernier octet de données renferme des zéros dans un à sept des bits de poids fort.

REPONSE :

- Les champs de la réponse sont décrits dans le paragraphe Interrogation.

Remarque

La demande Forcer une seule sortie n'est pas une commande de forçage permanente. La sortie spécifiée dans cette demande ne sera forcée à la valeur indiquée qu'au début d'un seul cycle de l'application Série 90.

MESSAGE (16) : **PREDEFINIR PLUSIEURS REGISTRES (ECRITURE)**

FORMAT :

Adresse	Fonct. 16	No. registre de départ	Nombre de registres	Nombre d'octets	Données	Contrôle d'erreurs

Interrogation

Adresse	Fonct. 16	No. registre de départ	Nombre de registres	Contrôle d'erreurs

Réponse normale

INTERROGATION :

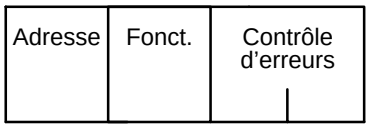
- L'**adresse** 0 indique une demande de diffusion. Toutes les stations esclaves traitent les demandes de diffusion mais sans renvoyer de réponse.
- Le **code de la fonction** est 16.
- Le **numéro du registre de départ** occupe deux octets et peut prendre n'importe quelle valeur, du moment qu'elle est inférieure au numéro du registre le plus haut disponible dans la CPU Série 90 connectée. Le numéro du registre de départ est égal au numéro du premier registre prédéfini par cette demande moins 1.
- La valeur **nombre de registres** occupe deux octets. Elle doit être comprise entre 1 et 125 inclus. La somme des valeurs Numéro du registre de départ et Nombre de registres doit être inférieure ou égale au numéro du registre le plus haut disponible dans la CPU Série 90 connectée. L'octet de poids fort des champs Numéro du registre de départ et Nombre de registres est transmis en premier, suivi de l'octet de poids faible de ces deux champs.
- Le champ **nombre d'octets** occupe un seul octet. Il doit renfermer un nombre binaire compris entre 2 et 250 inclus. C'est le nombre d'octets dans le champ de données de la demande Prédéfinir plusieurs registres. Le nombre d'octets est égal à deux fois le nombre de registres.
- Les registres sont retournés par numéro d'ordre dans le champ des **données** ; le registre ayant le plus petit numéro dans les deux premiers octets et celui ayant le plus grand numéro dans les deux derniers octets du champ de données. Le numéro du premier registre dans le champ de données correspond au numéro du registre de départ plus 1. L'octet de poids fort de chaque registre est transmis avant l'octet de poids faible.

REPONSE :

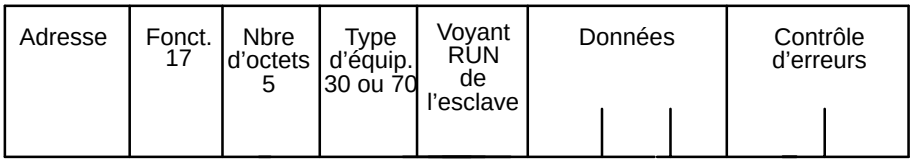
- Les champs de la réponse sont décrits dans le paragraphe Interrogation.

MESSAGE (17) : **RETOURNER TYPE D'EQUIPEMENT**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

L'interrogation Retourner Type d'équipement est transmise par l'équipement maître à l'équipement esclave pour connaître le type d'automate programmable industriel ou ordinateur utilisé.

- L'**adresse** 0 n'est pas autorisée dans la mesure où ce message ne peut pas être une demande de diffusion.
- Le **code de la fonction** est 17.

REPONSE :

- Le champ **nombre d'octets** occupe un octet et est égal à 5.
- Le champ **type d'équipement** occupe un octet et indique 30 si c'est un API Série 90–30 ou 70 si c'est un API Série 90–70.
- Le champ **voyant Run de l'esclave** occupe un octet. Il indique OFFH si la CPU Série 90 est en mode RUN et 0 si elle n'est pas en mode RUN.
- Le champ **données** renferme trois octets.

  Réponse possible pour l'API Série 90–30.

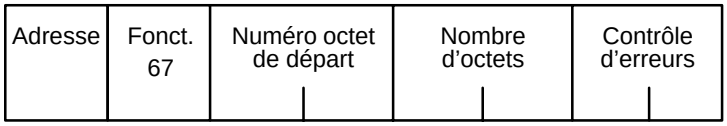
  35 00 00 pour CPU331
  36 00 00 pour CPU341

  Réponses possibles pour les API Série 90–70.

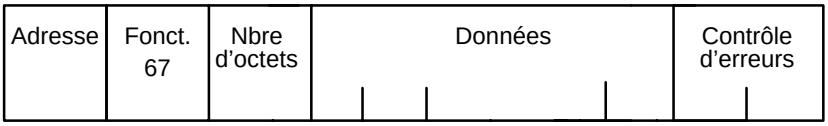
  31 00 00 pour CPU731
  32 00 00 pour CPU732
  71 00 00 pour CPU771
  72 00 00 pour CPU772
  81 00 00 pour CPU781
  82 00 00 pour CPU782
  88 00 00 pour CPU788
  89 00 00 pour CPU789
  92 00 00 pour CPU914
  94 00 00 pour CPU924

MESSAGE (67) : **LIRE MEMOIRE DE LA ZONE DE TRAVAIL**

FORMAT :



Interrogation



Réponse normale

INTERROGATION :

- L'**adresse** 0 n'est pas autorisée dans la mesure où ce message ne peut pas être une demande de diffusion.
- Le **code de la fonction** est 67.
- Le **numéro de l'octet de départ** occupe deux octets et peut prendre n'importe quelle valeur, du moment qu'elle est inférieure au numéro de l'adresse mémoire de la zone de travail le plus haut disponible dans la CPU Série 90 connectée, tel qu'indiqué dans le tableau ci–après. Le numéro de l'octet de départ est égal à l'adresse du premier octet de mémoire de la zone de travail retourné dans la réponse normale à cette demande.
- La valeur **nombre d'octets** occupe deux octets. Elle spécifie le nombre d'adresses mémoire (octets) de la zone de travail retourné dans la réponse normale. La somme des valeurs Numéro de l'octet de départ et Nombre d'octets doit être inférieure à 2 plus le plus haut numéro d'adresse mémoire de la zone de travail disponible dans la CPU Série 90 connectée. L'octet de poids fort des champs Numéro de l'octet de départ et Nombre d'octets est transmis en premier, suivi de l'octet de poids faible de ces deux champs.

REPONSE :

- Le **nombre d'octets** est un nombre binaire compris entre 0 et 255 (0 = 256). C'est le nombre d'octets figurant dans le champ de données de la réponse normale.
- Le champ **données** renferme le contenu de la mémoire de zone de travail demandé dans l'interrogation. Le contenu de l'octet de mémoire de la zone de travail dont l'adresse correspond au numéro de l'octet de départ est transmis dans le premier octet du champ de données. Le contenu de l'octet de mémoire de travail dont l'adresse correspond à la somme des valeurs Numéro de l'octet de départ et Nombre d'octets moins 1 est transmis dans le dernier octet du champ de données.

Zone de travail RTU

La zone de travail est entièrement mise à jour chaque fois qu'une demande READ (Lecture) externe est reçue par le RTU. Toutes les adresses de la zone de travail sont de type *lecture seule*. La zone de travail est un type de mémoire orienté Octet.

Tableau 7-13. Affectation de la mémoire de zone de travail RTU

Adresse	Identificateur de champ	Bits							
		7	6	5	4	3	2	1	0
00	Etat Run CPU	0	0	0	0	Voir Notes (1)			
01	Etat Commande CPU	Configuration binaire identique à SP(00)							
02 03	Type CPU	Majeur(2a) (en hexadécimal) Mineur(2b) (en hexadécimal)							
04–0B	ID CPU SNP	7 caractères ASCII + caractère de fin = (00h)							
0C 0D	No. Révision Microprogramme CPU	Majeur Mineur							
0E 0F	No. Révision Microprogramme CMM	Majeur Mineur							
10–11	Réservées	(00h)							
12	ID Type d'abonné	(90–70 : 0Ch ; 90–30 : 0Dh)							
13–15	Réservées	(00h)							
16	Adresse de la station RTU	1–247 (en décimal)							
17	Réservée	(00h)							
18–33	Tailles des types de mémoire	Voir Note (3)							
18–1B	Mémoire des registres	Taille %R							
1C–1F	Table des entrées analogiques	Taille %AI							
20–23	Table des sorties analogiques	Taille %AQ							
24–27	Table des entrées	Taille %I							
28–2B	Table des sorties	Taille %Q							
2C–2F	Mémoire logique interne	Taille %M							
30–33	Code Programme Utilisateur	Voir Note (4)							
34–FF	Réservées	(00h)							

Notes relatives aux affectations de la mémoire de zone de travail

1 0000 = Run_Validé
0001 = Run_Invalidé
0010 = Stoppé
0011 = Stoppé_En_défaut
0100 = Arrrêté
0101 = Suspendu
0110 = Stoppé_ES_Validées

2a Types majeurs des CPU :
12 (0Ch) CPU API Série 90–70
16 (10h) CPU API Série 90–30

2b Types mineurs des CPU Série 90–70 :
31 (1Fh) CPU 731
32 (20h) CPU 732
71 (47h) CPU 771
72 (48h) CPU 772
81 (51h) CPU 781
82 (52h) CPU 782
88 (58h) CPU 788
89 (59h) CPU 789
92 (5ch) CPU 914
94 (5eh) CPU 924

Sous–types Série 90–30 :
35 (23h) CPU 331
36 (24h) CPU 341

3 Octets 18h–33h de la zone de travail :

Octets	Longueur de la mémoire		Taille retournée en
18–1B	%R	Mémoire des registres	Mots
1C–1F	%AI	Table des entrées analogiques	Mots
20–23	%AQ	Table des sorties analogiques	Mots
24–27	%I	Table des entrées	Points (Bits)
28–2B	%Q	Table des sorties	Points (Bits)
2C–2F	%M	Mémoire logique interne	Points (Bits)
30–33		Code Programme Utilisateur	Octets

Remarque

La longueur en hexadécimal de chaque type de mémoire est codée sur quatre octets dont le mot de poids fort est réservé pour un usage ultérieur. Par exemple, la taille par défaut de 1024 mots (0400h) de la mémoire des registres du 731 serait retournée sous le format suivant :

Mot	Poids faible		Poids fort	
Octet SP	18	19	1A	1B
contient	00	04	00	00

4 Espace de mémoire programme occupé par le programme. S'affiche également dans l'écran Mémoire utilisée de l'API Logicmaster 90 dans le champ Prog Utilisateur.

2.10.1. Erreurs de communication

Les erreurs de communication de la liaison série se répartissent en trois groupes :

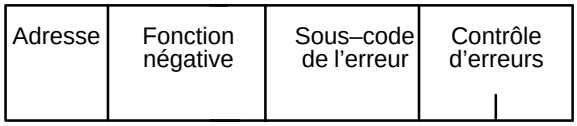
- Message d'interrogation invalide
- Temporisations de la liaison série
- Transaction invalide

2.10.2. Message d'interrogation invalide

Lorsque le module de communication reçoit une interrogation qui lui est adressée mais qu'il ne peut pas traiter, il envoie une des réponses d'erreur suivantes :

	Sous–code
Code de la fonction invalide	1
Champ Adresse invalide	2
Champ Données invalide	3
Echec du traitement de l'interrogation	4

Le format d'une réponse d'erreur à une interrogation est le suivant :



L'adresse indique l'adresse fournie dans la demande d'origine. Le code des fonctions négatives est égal à la somme du code de la fonction plus 128. Le sous–code de l'erreur est 1, 2, 3 ou 4. La valeur du sous–code indique la raison pour laquelle l'interrogation n'a pas pu être traitée.

2.10.2.1. Réponse d'erreur (1) Code de la fonction invalide

La réponse d'erreur ayant le sous–code 1 correspond à Code de la fonction invalide. Cette réponse est envoyée par un équipement esclave lorsqu'il recoit un message d'interrogation dont le code de la fonction n'est pas égal à 1–8, 15, 16, 17 ou 67.

2.10.2.2. Réponse d'erreur (2) Adresse invalide

La réponse d'erreur ayant le sous–code 2 correspond à Adresse invalide. Cette réponse est envoyée dans les cas suivants :

1. Les champs Numéro du point de départ et Nombre de points spécifient des points de sortie ou d'entrée qui ne sont pas disponibles dans la CPU Série 90 connectée (réponse retournée pour codes des fonctions 1, 2, 15).
2. Les champs Numéro du registre de départ et Nombre de registres spécifient des registres qui ne sont pas disponibles dans la CPU Série 90 connectée (réponse retournée pour codes des fonctions 4, 16).
3. Les champs Numéro de l'entrée analogique de départ et Nombre d'entrées analogiques spécifient des entrées analogiques qui ne sont pas disponibles dans la CPU Série 90 connectée (réponse retournée pour code de la fonction 3).
4. Le champ Numéro du point spécifie un point de sortie qui n'est pas disponible dans la CPU Série 90 connectée (réponse retournée pour code de la fonction 5).

5. Le champ Numéro du registre spécifie un registre qui n'est pas disponible dans la CPU Série 90 connectée (réponse retournée pour code de la fonction 6).

6. Le champ Numéro de l'entrée analogique spécifie une entrée analogique qui n'est pas disponible dans la CPU Série 90 connectée (réponse retournée pour code de la fonction 3).

7. Le code de diagnostic n'est pas égal à 0, 1 ou 4 (réponse retournée pour code de la fonction 3).

8. Les champs Numéro de l'octet de départ et Nombre d'octets spécifient une adresse mémoire de la zone de travail qui n'est pas disponible dans la CPU Série 90 connectée (réponse retournée pour code de la fonction 67).

2.10.2.3. Réponse d'erreur (3) Valeur de donnée invalide

La réponse d'erreur ayant le sous–code 3 correspond à Valeur de donnée invalide. Cette réponse est envoyée dans les cas suivants :

Le premier octet du champ de données n'est pas égal à 0 ou 255 (FFh) ou le second octet du champ de données n'est pas égal à 0 pour la demande Forcer une seule sortie (code de la fonction 5) ou Provoquer redémarrage des communications (code de la fonction 8, code de diagnostic 1). Les deux octets du champ de données ne sont pas tous deux égal à 0 pour la demande Forcer mode Réception seulement (code de la fonction 8, code de diagnostic 4).

Cette réponse est également transmise lorsque la longueur des données spécifiée par le champ Adresse mémoire est supérieure à celle des données reçues.

2.10.2.4. Réponse d'erreur (4) Echec du traitement de l'interrogation

La réponse d'erreur ayant le sous–code 4 correspond à Echec du traitement de l'interrogation. Cette réponse est envoyée par un équipement RTU lorsqu'il reçoit correctement une interrogation mais que la communication entre la CPU Série 90 associée et le module CMM se solde par un échec.

2.10.3. Temporisations de la liaison série

La temporisation d'un équipement RTU expire uniquement lorsqu'une interruption dans le flux des données supérieure au temps de transmission de 3 caractères survient lors de la réception d'un message. Dans ce cas, le message est considéré comme terminé et l'équipement RTU ne répondra pas à l'équipement maître. Certaines particularités en matière de synchronisation, dues aux caractéristiques de l'équipement esclave, doivent être prises en compte par l'équipement maître.

Après avoir envoyé un message d'interrogation, l'équipement maître doit attendre environ 500 millisecondes avant de considérer que l'équipement esclave n'a pas répondu.

2.10.4. Transactions invalides

Si une erreur n'entrant ni dans la catégorie d'un message d'interrogation invalide, ni dans celle d'une temporisation de la liaison série survient, on considère qu'elle correspond à une transaction invalide. Les types d'erreurs classés dans la catégorie des transactions invalides comprennent :

- CRC erroné.
- Longueur des données spécifiée par le champ Adresse mémoire supérieure à celle des données reçues.
- Erreurs de perte d'encadrement de trame ou de séquence.
- Erreurs de parité.

Si une erreur relevant de cette catégorie survient lors de la réception d'un message par le module CMM, l'équipement RTU ne retourne pas de message d'erreur. L'équipement RTU esclave traite le message entrant comme s'il ne lui était pas adressé.

3. PROTOCOLE SNP

Le protocole SNP est décrit de façon simple pour les utilisateurs des mises en oeuvre SNP maître et esclave sur le module CMM. Parmi les sujets abordés figurent le flux du protocole, le mode maître–esclave, la synchronisation du protocole et le principe des datagrammes. Le protocole SNP transfère les données en utilisant le format de données série asynchrone. Pour plus d'informations sur la transmission série et le format asynchrone, voir Chapitre 2, *Série 90 : notions de communication série*, § 2, *L'interface série*.

Ce paragraphe traite des sujets suivants :

- Présentation du protocole SNP
- Principe du mode SNP maître–esclave
- Temporisateurs SNP
- Datagrammes SNP

3.1. PRÉSENTATION DU PROTOCOLE SNP

Le protocole SNP est un protocole de communication série propriétaire développé pour permettre aux API de communiquer avec d'autres équipements. Le protocole est un ensemble de règles qui établit et maintient une liaison de communication entre l'équipement SNP maître et le(s) équipement(s) SNP esclaves.

SNP est un protocole half–duplex qui utilise les interfaces électriques RS–485 (version étendue de RS–422) et RS–232. L'équipement SNP esclave intégré dans la CPU des API Série 90 utilise l'interface RS–485. Le module CMM fournit à la fois les interfaces RS–485 et RS–232. (Pour plus d'informations sur l'interface électrique du module CMM, voir Chapitre 3). Les équipements SNP peuvent être configurés en mode point–à–point direct (un équipement maître et un équipement esclave) ou multipoint (un équipement maître et plusieurs équipements esclaves). Un seul maître peut figurer dans une configuration multipoint. Lorsque plusieurs équipements se partagent une liaison multipoint, chacun d'eux est identifié par un ID SNP unique.

3.2. PRINCIPE DU MODE SNP MAÎTRE–ESCLAVE

SNP est un protocole maître–esclave dans lequel l'équipement maître établit toutes les communications et l'équipement esclave répond aux demandes du maître. SNP ne fournit pas la fonction égal à égal.

Chaque API Série 90 renferme un équipement SNP esclave interne sur le port série console de la CPU. (Le port série de la CPU est fourni sur la carte CPU dans les API Série 90–70 et Série 90–20, et sur l'alimentation dans les API Série 90–30).

Le gestionnaire SNP DOS est un équipement SNP maître ; les applications PC qui utilisent ce gestionnaire peuvent communiquer avec des API Série 90 par une liaison SNP. Logicmaster 90 utilise un SNP maître pour communiquer, et pour configurer et programmer les API Série 90. Des équipements SNP maîtres ont de plus été développés par de nombreux autres fabricants pour communiquer avec des API Série 90.

Chaque port série fourni par le module CMM dans un API Série 90 peut être configuré comme équipement SNP maître ou esclave (en plus de l'équipement SNP esclave interne sur le port série de la CPU). Les équipements SNP dans le module CMM permettent d'échanger des données avec d'autres équipements SNP. Les équipements SNP maître et esclaves sur le module CMM ne supportent pas les fonctions de programmation ou de configuration de l'API.

3.2.1. Format des messages

Il existe trois types de messages SNP : les messages de type Connexion (Connexion, Réponse de connexion, et Mettre à jour le datagramme en temps réel), dont la longueur est toujours de 24 octets ; les messages Boîte aux lettres dont la longueur est toujours de 40 octets ; et les messages Tampon dont la longueur varie suivant le volume de données. Le contenu de chaque message varie suivant le type de message et les données transférées. Le protocole SNP renvoie également un accusé de réception de 2 octets après chaque message Boîte aux lettres ou Tampon pour confirmer sa réception.

3.2.1.1. Etablissement d'une communication SNP

Une communication SNP se décompose en deux phases : une session de communication doit d'abord être établie. Sur le module CMM, elle est établie par les commandes Connexion ou Connexion longue. Une fois la communication établie, les données peuvent être transférées entre le maître et l'esclave. Sur le module CMM, elles sont transférées par le jeu de commandes SNP déportées. (Pour plus d'informations sur toutes les commandes SNP disponibles sur le module CMM, voir Chapitre 6, Service SNP).

Le maître établit une session de communication comme suit : il envoie une séquence Break à tous les équipements sur la liaison série. La séquence Break consiste à maintenir le signal Emettre à l'état Space (signal à 0) pendant une durée supérieure au temps de transmission de deux caractères à la vitesse sélectionnée. (Sur le module CMM, la séquence Break doit correspondre au temps de transmission de trois caractères et ne peut pas être inférieure à 5 ms ; Si le paramètre Délai TurnA a été défini, la séquence Break est fixée à 400 ms). Le module CMM envoie une séquence Break dans la commande Connexion, Connexion longue et Mettre le datagramme à jour.

A réception d'une séquence Break, tous les équipements esclaves arrêtent immédiatement les communications SNP en cours et se préparent à recevoir un message de connexion de l'équipement maître.

A la fin de la séquence Break, l'équipement maître attend pendant l'intervalle de temps spécifié dans T4 avant d'envoyer un message de connexion. Le message de connexion renferme l'ID SNP de l'équipement esclave avec lequel l'équipement maître désire établir la session de communication. Seul l'équipement esclave spécifié répondra à l'équipement maître par un message Réponse de connexion ; la session de communication est alors établie entre l'équipement maître et l'équipement esclave spécifié. Tous les autres équipements esclaves sur la liaison série attendent une nouvelle séquence Break.

L'ID SNP de l'équipement SNP esclave intégré dans la CPU de tous les API Série 90 est défini par le logiciel de configuration Logicmaster 90. Pour un équipement SNP esclave sur un module CMM, l'ID SNP prend par défaut le même ID SNP que celui utilisé par la CPU de l'API. Cette valeur par défaut peut être modifiée par la commande COMREQ "Modifier ID SNP". (Pour plus d'informations sur cette commande, voir Chapitre 6, Service SNP).

Le flux des messages SNP permettant d'établir une session de communication SNP est illustré ci–dessous :

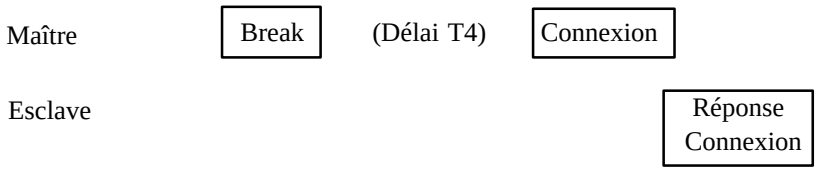


Figure 7-10. Flux des messages : Etablir une session de communication

Une fois établie, la session de communication reste active jusqu'à ce que l'équipement maître envoie une autre séquence Break, ou qu'une temporisation ou autre erreur bloquante survienne, soit dans l'équipement maître, soit dans l'équipement esclave.

3.2.1.2. Demande et réponse

Une fois que la communication a été établie, l'équipement maître peut envoyer une demande à l'équipement esclave qui doit obligatoirement répondre. Une demande ou une réponse consiste en un message Boîte aux lettres, éventuellement suivi d'un ou de plusieurs messages Tampon. L'équipement maître, tout comme l'équipement esclave, procède à un contrôle d'erreurs sur tous les messages reçus et renvoie un accusé de réception à l'expéditeur du message.

Le flux des messages SNP pour les demandes et les réponses SNP est illustré ci–dessous :

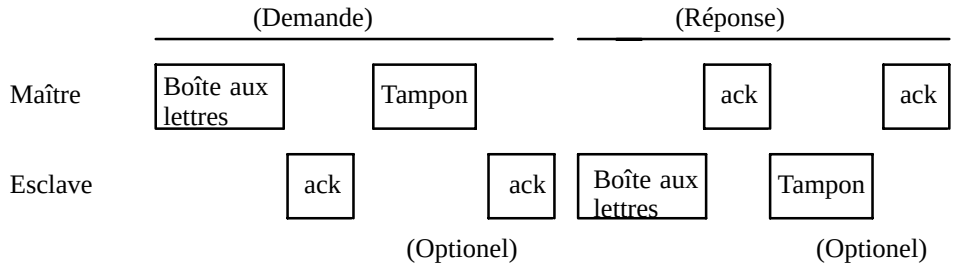


Figure 7-11. Flux des messages : Demande et réponse

Chaque message SNP doit être acquitté. Si une fois le message envoyé, aucun accusé de réception ne survient dans l'intervalle de temps défini par T2, la communication SNP est annulée. L'équipement maître doit établir une nouvelle session de communication.

Toutes les demandes SNP doivent recevoir une réponse. Une fois que l'équipement maître a complété une demande (tous les messages de demande ont été envoyés et acquittés), si le premier message de réponse n'est pas reçu dans l'intervalle de temps défini par T5', la communication SNP est annulée. L'équipement maître doit établir une nouvelle session de communication.

Lorsqu'une demande ou une réponse renferme un ou plusieurs messages Tampon, le message Tampon suivant doit suivre le précédent chronologiquement. Si un message Tampon attendu n'est pas reçu dans l'intervalle de temps défini par T5'', la communication SNP est annulée. L'équipement maître doit établir une nouvelle session de communication.

Pour plus d'informations sur les temporisateurs SNP, voir paragraphe Temporisateurs SNP, ci–après.

Même si aucune erreur de transmission ou d'intégrité ne survient, l'équipement esclave peut ne pas réussir à exécuter l'action demandée par l'équipement maître. Dans ce cas, il signale l'échec de la demande en retournant les codes d'erreurs majeures et mineures appropriés à l'équipement maître dans un message Boîte aux lettres.

3.2.1.3. Essais de retransmission et reprise sur incident

Dès qu'un message SNP est reçu, il est examiné pour voir s'il ne comporte pas d'erreurs. Si une erreur de tranmission ou d'intégrité du message est détectée, un accusé de réception négatif est retourné, accompagné d'un code d'erreur.

Si l'accusé de réception négatif indique une erreur non bloquante, l'expéditeur essaie de retransmettre le message. L'équipement SNP maître autorise deux essais de retransmission maximum, quelque soit le type de message. Si le message n'est pas correctement envoyé ou reçu après deux essais, l'équipement maître annule la communication SNP et doit établir une nouvelle session de communication.

Si l'accusé de réception négatif indique une erreur bloquante, le maître et l'esclave annulent tous deux immédiatement la communication. L'équipement maître doit établir une nouvelle session de communication.

3.2.1.4. Erreurs CMM bloquantes

Certaines erreurs locales signalent des erreurs bloquantes sur le module CMM. Dans ce cas, le module CMM annule la communication SNP de son côté. Une nouvelle session de communication doit être établie. Les erreurs locales bloquantes sont signalées par le code d'erreur majeure 12 (0Ch) et les codes d'erreurs mineures 32 (20h) et plus. (La liste complète et la description de tous les codes d'erreurs majeures et mineures figurent dans le Chapitre 6, Service SNP).

3.3. TEMPORISATEURS SNP

Plusieurs temporisateurs sont définis pour gérer le protocole SNP.

Des temporisateurs réglables sont fournis dans la mesure où des équipements SNP différents peuvent coexister sur une liaison série donnée, avec éventuellement des processeurs et des fonctions de traitement qui diffèrent. Le protocole SNP fournit plusieurs temporisateurs dont les valeurs peuvent être spécifiées par l'utilisateur pour que la liaison fonctionne aussi efficacement que possible dans une situation donnée.

La valeur par défaut de chaque temporisateur SNP est sélectionnée par le paramètre de configuration du module CMM "Temps limit". Les quatre choix "Temps limit" possibles fournissent un jeu de valeurs pour tous les temporisateurs SNP. Les valeurs par défaut des temporisateurs SNP correspondant à chaque choix sont indiquées dan le Tableau 3.4, Temporisations SNP. Des temporisations différentes des valeurs par défaut peuvent être fournies par la commande SNP Connexion longue. Cette commande génère un message Sélectionner les paramètres SNP pour permettre à l'équipement maître de négocier de nouvelles valeurs de temporisation avec les équipements esclaves.

Tous les temporisateurs reviennent à leur valeur de configuration par défaut lorsqu'une nouvelle demande de connexion est transmise (commande de connexion). Les valeurs de temporisation autres que celles par défaut doivent être rétablies après chaque nouvelle demande de connexion. Les valeurs des temporisateurs SNP prédéfinies ne sont pas modifiées par la demande Mettre à jour le datagramme en temps réel.

Chaque temporisateur SNP est décrit ci–dessous. Le délai de retournement du modem et le délai de transmission ne sont *pas* inclus dans les temporisations SNP, sauf spécification expresse. Le délai de retournement du modem et le délai de transmission sont décrits à la suite des temporisateurs.

T1 **Temps de retournement minimum**. (maître et esclave) C'est l'intervalle de temps minimal requis par l'équipement entre une réception et la transmission qui suit. Il permet à la ligne série half–duplex de changer de sens. Ce laps de temps est généralement utilisé comme délai entre la réception d'un message et l'envoi de son accusé de réception, ou entre la réception d'un accusé de réception et l'envoi du prochain message. Il doit s'écouler complètement avant l'envoi d'un message ou d'un accusé de réception.

La valeur du temporisateur T1 est définie par défaut à 5 ms pour tous les choix du paramètre de configuration du module CMM "Temps limit". Les valeurs du temporisateur T1 sont toujours échangées entre les équipements maître et esclave dans la demande de connexion et les messages de réponse ; chaque équipement utilise la valeur T1 obtenue de l'autre.

T2 **Temporisation de l'accusé de réception.** (maître et esclave) C'est l'intervalle de temps maximal autorisé pour recevoir un accusé de réception après l'envoi d'un message. Si l'accusé de réception n'est pas reçu dans l'intervalle de temps défini par T2, l'équipement SNP (maître ou esclave) annule la communication SNP.

La valeur par défaut du temporisateur T2 est définie par le paramètre de configuration "Temps limit" ; elle varie entre 0,5 s et 2 s. Les valeurs du temporisateur T2 peuvent éventuellement être négociées entre les équipements maître et esclave via le message Sélectionner les paramètres SNP (fourni par la commande Connexion longue). Les équipements maître et esclave utilisent tous deux la plus grande valeur T2 fournie par l'un ou l'autre équipement.

(Chaque équipement ajoute les valeurs du délai de retournement du modem et du délai de transmission négociées séparément à la valeur T2 négociée avant utilisation. Vous ne devez pas inclure ces valeurs dans la valeur T2 configurée ou programmée).

T3 **Temps de maintien de la liaison.** (maître seulement) C'est l'intervalle de temps maximal pendant lequel l'équipement maître attendra sans qu'il n'y ait d'activité sur la ligne série. Il démarre dès la fin d'une réponse de connexion ou autre réponse SNP. Si l'équipement maître n'a envoyé aucun message SNP dans l'intervalle de temps défini par T3, il enverra un message de demande de renseignement ou Etat court à l'équipement esclave connecté pour éviter que la temporisation T3' n'arrive à expiration sur l'équipement esclave.

Le temporisateur T3 est toujours défini à la valeur T3' courante moins le temps requis pour transférer 80 caractères à la vitesse sélectionnée, moins tout délai de retournement du modem négocié. La valeur T3 est recalculée chaque fois qu'une nouvelle valeur T3' est établie. Si le temporisateur T3' est désactivé, le temporisateur T3 l'est également.

T3' **Temporisation d'inactivité de la liaison.** (esclave seulement) C'est l'intervalle de temps maximal pendant lequel l'équipement esclave attendra une activité sur la ligne série. Il démarre dès la fin d'une réponse SNP et est annulé dès réception d'une nouvelle demande de la part de l'équipement maître. Si aucune activité ne se manifeste dans l'intervalle de temps défini par T3', l'équipement esclave annule la communication SNP.

La valeur par défaut du temporisateur T3' est définie par le paramètre de configuration du module CMM "Temps limit" ; elle varie entre 0 (désactivé) et 10 s. La valeur du temporisateur T3' peut éventuellement être négociée entre les équipements maître et esclave via le message Sélectionner les paramètres SNP (fourni par la commande Connexion longue). Les équipements maître et esclave utilisent tous deux la plus grande valeur T3' fournie par l'un ou l'autre équipement.

(Chaque équipement ajoute la valeur du délai de retournement du modem négociée séparément à la valeur T3' négociée avant utilisation. Vous ne devez pas inclure cette valeur dans la valeur T3' configurée ou programmée).

T3'' **Temporisation de connexion inactive.** (esclave seulement) C'est un cas particulier d'utilisation du temporisateur T3' par l'équipement esclave uniquement, immédiatement après une nouvelle demande/réponse de connexion. Il laisse à l'équipement maître suffisamment de temps pour négocier les nouvelles valeurs de temporisation en envoyant une demande Sélectionner les paramètres SNP à l'équipement esclave. Cette valeur de temporisation spéciale n'est utilisée que jusqu'à réception du premier message qui suit la commande de connexion ; après quoi, la valeur par défaut ou négociée T3' est utilisée.

La valeur du temporisateur T3'' est toujours égale à la valeur par défaut de T3' plus un délai de transmission fixe de 65 s correspondant aux cas les plus défavorables. Si le temporisateur T3' est désactivé, le temporisateur T3'' l'est également.

T4 **Temps de traitement d'un Break.** (maître seulement) C'est l'intervalle de temps minimal pendant lequel l'équipement maître doit attendre à la fin d'une séquence Break avant d'envoyer un message de connexion ou Mettre à jour le datagramme en temps réel. Cet intervalle de temps permet à l'équipement esclave d'activer ses circuits de réception et de se préparer à accepter les messages entrants. Il n'est utilisé que par l'équipement maître. Ce temporisateur est utilisé dans les commandes de connexion, connexion longue et Mettre à jour le datagramme en temps réel.

La valeur par défaut du temporisateur T4 est définie à 50 ms pour tous les choix du paramètre de configuration du module CMM "Temps limit". Cette valeur peut éventuellement être modifiée par la commande de connexion longue.

T5 Temps de traitement maximal. (esclave seulement) C'est l'intervalle de temps maximal dont bénéficie l'équipement esclave pour préparer un message de réponse. Il démarre lorsque le dernier accusé de réception de la demande est envoyé à l'équipement maître. Si le message de réponse n'est pas prêt à être envoyé dans l'intervalle de temps défini par T5, l'équipement esclave enverra un message Maintien de la liaison à l'équipement maître pour éviter que la temporisation T5' n'arrive à expiration sur l'équipement maître. (En règle générale, le temporisateur T5 garde une trace du temps pendant lequel l'équipement SNP esclave attend le processeur des demandes de service).

La valeur par défaut du temporisateur T5 est définie par le paramètre de configuration du module CMM "Temps limit" ; elle varie entre 0 (désactivé) et 5 s. La valeur T5 utilisée par l'équipement maître peut être modifiée par la commande de connexion longue. La valeur du temporisateur T5 n'est pas négociée entre les équipements maître et esclaves ; l'équipement esclave utilise toujours la valeur définie pour T5. Le temporisateur T5 est désactivé chaque fois que le temporisateur T2 (Temporisation de l'accusé de réception) est désactivé.

T5' Temporisation du traitement. (maître seulement) C'est l'intervalle de temps maximal pendant lequel l'équipement maître attendra que l'équipement esclave retourne le message de réponse attendu. Il démarre à réception du dernier accusé de réception de la demande en provenance de l'équipement esclave. Il est redémarré à réception d'un message Maintien de la liaison de la part de l'équipement esclave. Si la réponse n'est pas reçue dans l'intervalle de temps défini par T5', la communication SNP est annulée.

Le temporisateur T5' est toujours défini à la valeur T5 courante plus le temps requis pour transférer 80 caractères à la vitesse sélectionnée, plus tout délai de retournement du modem négocié. La valeur T5' est recalculée chaque fois qu'une nouvelle valeur T5 est établie. Si le temporisateur T2 ou T5 est désactivé, le temporisateur T5' l'est également.

T5'' Temporisation du message Tampon. (maître et esclave) C'est l'intervalle de temps maximal pendant lequel une station attendra que l'autre station lui envoie le message Tampon attendu dans une demande ou une réponse SNP. Il démarre après qu'un accusé de réception ait été envoyé lorsqu'un autre message Tampon est attendu. Si le message Tampon n'est pas reçu dans l'intervalle de temps défini par T5'', la communication SNP est annulée.

Le temporisateur T5'' est toujours défini à la valeur T5' courante plus le délai de transmission plus deux fois le temps requis pour transférer le message Tampon suivant à la vitesse sélectionnée. La valeur T5'' est recalculée chaque fois qu'une message Tampon est attendu. Si le temporisateur T2 ou T5 est désactivé, le temporisateur T5'' l'est également.

Le délai de retournement du modem et le délai de transmission qui s'ajoutent à certains temporisateurs SNP, sont décrits ci–dessous :

Délai de retournement du modem : (maître et esclave) C'est l'intervalle de temps requis par les modems intermédiaires pour inverser le sens de la liaison.

La valeur par défaut du délai de retournement du modem est spécifiée par le paramètre de configuration du module CMM "Délai TurnA" ; elle varie entre 0 (Sans) et 500 ms. La valeur du délai de retournement du modem peut éventuellement être négociée entre les équipements maître et esclave via le message Sélectionner les paramètres SNP (fourni par la commande Connexion longue). Les équipements maître et esclave utilisent tous deux la plus grande valeur de délai de retournement du modem fournie par l'un ou l'autre équipement.

Délai de transmission : (maître et esclave) C'est l'intervalle de temps requis pour que les données soient transférées de l'équipement initiateur à l'équipement répondeur. (Ce temps est également appelé "temps de câble"). Il reflète les délais inhabituels dans la voie de transmission des données, lorsque les liaisons satellites sont utilisées pour transférer les données par exemple.

Le délai de transmission est défini à 0 ms par défaut pour tous les choix du paramètre de configuration du module CMM "Temps limit". La valeur du délai de transmission peut éventuellement être négociée entre les équipements

maître et esclave via le message Sélectionner les paramètres SNP (fourni par la commande Connexion longue). Les équipements maître et esclave utilisent tous deux la plus grande valeur de délai de transmission fournie par l'un ou l'autre équipement.

3.4. DATAGRAMMES SNP

Les demandes SNP Lire et Ecrire permettent à l'équipement maître de lire ou d'écrire directement dans la mémoire de l'équipement esclave. Chaque demande de lecture ou d'écriture ne peut porter que sur un seul type de mémoire à la fois ; plusieurs demandes sont nécessaires pour accéder à différents types de mémoires. SNP fournit également la possibilité de lire plusieurs types de mémoires d'un équipement esclave à partir d'une seule demande. La référence des mémoires mixtes d'un équipement esclave est appelée un Datagramme. Les datagrammes permettent d'accéder à des types de mémoires mixtes plus rapidement que si l'on procédait à des lectures successives de chaque type de mémoire.

L'accès à un datagramme diffère d'un accès direct à la mémoire dans la mesure où les références de mémoires doivent être préalablement établies. Une fois le datagramme établi, les données spécifiées peuvent être extraites de la mémoire de l'équipement esclave par une simple commande de lecture du datagramme. Le datagramme peut être annulé à tout instant pour libérer des ressources dans la CPU de l'API.

3.4.1. Etablissement d'un datagramme

L'équipement SNP maître établit un datagramme en spécifiant une ou plusieurs zones de mémoire dans l'équipement esclave. Chaque zone de mémoire est définie par un format Point, qui spécifie le type de mémoire, l'adresse mémoire et le nombre d'éléments de mémoire correspondant au type de mémoire de l'équipement esclave. L'équipement SNP maître dans le module CMM peut définir jusqu'à 32 formats Point lors de la création d'un datagramme.

Lors de l'établissement du datagramme, l'équipement esclave alloue des ressources à la CPU de l'API pour regrouper les diverses zones de mémoire dans une zone de données contiguë qui pourra être transmise à l'équipement maître sur sa demande. Lorsque le datagramme est établi avec succès, l'équipement esclave retourne un ID Datagramme à l'équipement maître. L'ID Datagramme est un octet unique que l'équipement maître devra spécifier pour identifier le datagramme dans toutes ses futures demandes.

Tous les types de mémoire de l'équipement esclave accessibles par les autres commandes de lecture peuvent être spécifiés dans un format Point, y compris la mémoire des tâches du programme principal (pour les équipements Série 90–70 esclaves seulement) et la mémoire des blocs du programme.

Les types de mémoire orientés Bit ne sont *pas* supportés dans les formats Points par les équipements Série 90–70 esclaves. Des restrictions significatives limitent l'utilisation des types de mémoire orientés Bit dans un format Point pour les équipements Série 90–30 ou Série 90–20 esclaves : la zone de mémoire définie par un format Point ne doit pas excéder un octet standard de mémoire dans l'équipement esclave (éléments #1 à #8, #9 à #16, ..., #(N*8)+1 à #(N*8)+8, ...). Dans la mesure où les types de mémoire orientés Bit peuvent également être référencés comme des types de mémoire orientés Octet, il est recommandé (et obligatoire pour les équipements Série 90–70 esclaves) d'utiliser un type de mémoire orienté Octet plutôt que Bit dans un format Point. Les types de mémoire orientés Octets dans un format Point ne sont soumis à aucune restriction. (Voir Chapitre 6, Services SNP, Tableau 6.1, Types de mémoire, unités de longueur et Gammes valides pour connaître tous les types de mémoire accessibles. Ce tableau indique également l'équivalent orienté Octet de chaque type de mémoire orienté Bit).

3.4.1.1. Extraction d'un datagramme

Les zones de mémoire de l'équipement esclave définies par un datagramme sont retournées à l'équipement maître lorsque celui–ci génère une demande Mettre à jour le datagramme ou Mettre à jour le datagramme en temps réel. L'équipement SNP maître dans le module CMM sauvegarde les données de la réponse dans la mémoire de l'API local spécifiée par la commande de mise à jour ; un espace de mémoire suffisant doit être spécifié pour contenir la réponse complète.

3.4.1.2. Mise à jour du datagramme en temps réel

Pour qu'une demande Mettre à jour le datagramme (demande Boîte aux lettres) aboutisse, une session de communication SNP doit avoir été préalablement établie entre l'équipement maître et esclave via une commande de connexion ou connexion longue. La demande Mettre à jour le datagramme en temps réel diffère de la demande Mettre à jour le datagramme dans la mesure où aucune communication préalable avec l'équipement esclave désiré n'est requise.

Mettre à jour le datagramme en temps réel est un service spécial qui permet à l'équipement maître d'établir une nouvelle communication avec l'équipement esclave désiré, de spécifier l'ID du datagramme précédemment établi, et d'extraire immédiatement le datagramme de cet équipement esclave. La demande de connexion préalable n'étant pas requise, le service Mettre à jour le datagramme en temps réel constitue le moyen le plus rapide pour interroger tous les équipements esclaves dans une configuration multipoint.

Le service Mettre à jour le datagramme en temps réel ne s'applique qu'aux datagrammes permanents. Voir dans le paragraphe Durée de vie d'un datagramme, ci–dessous, la différence entre un datagramme normal et un datagramme permanent.

Le flux des messages SNP pour le service Mettre à jour le datagramme en temps réel est illustré ci–dessous :

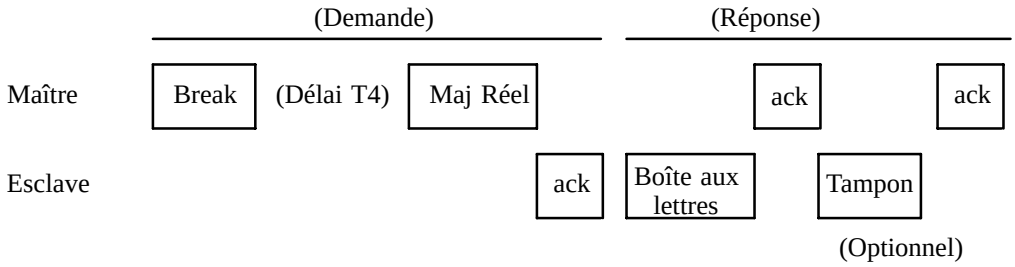


Figure 7-12. Flux des messages : Mettre à jour le datagramme en temps réel

Remarque

Toute communication SNP existante est annulée lorsque l'équipement maître envoie la demande Mettre à jour le datagramme en temps réel. Cette demande génère une séquence Break qui force tous les équipements esclaves sur la liaison série à arrêter les communications en cours et à attendre une demande de connexion ou de mise à jour du datagramme en temps réel. Comme la demande de connexion, la demande Mettre à jour le datagramme en temps réel spécifie un équipement esclave particulier. L'équipement concerné accuse réception de la demande Mettre à jour le datagramme en temps réel ; les autres équipements esclaves attendent la prochaine séquence Break.

3.4.1.3. Durée de vie d'un datagramme

Tout datagramme établi peut être annulé immédiatement via la demande Annuler le datagramme. Tous les datagrammes établis sont automatiquement annulés par la mise hors tension de l'équipement esclave. De plus, certains datagrammes sont automatiquement annulés lorsque la communication SNP sous laquelle ils ont été établis prend fin.

Les datagrammes normaux établis dans un équipement esclave CMM711 (Série 90–70) ou l'équipement esclave interne de la CPU d'un API Série 90 sont automatiquement annulés lorsque la communication SNP courante prend fin. Les datagrammes normaux établis dans un équipement esclave CMM311 (Série 90–30) se comportent comme des datagrammes permanents ; ils ne sont pas annulés à la fin de la communication SNP courante.

Tout datagramme (normal ou permanent) peut être annulé à la demande.

L'ID Datagramme spécifié dans une demande Annuler le datagramme permet d'identifier le datagramme à annuler. La valeur ID "–1" peut être utilisée pour spécifier tous les datagrammes établis du type indiqué (normal ou permanent).

Remarque

Cas particulier : lorsque les deux ports série d'un module CMM311 (Série 90–30) sont configurés comme des équipements SNP esclaves, l'ID "–1" dans une demande Annuler le datagramme reçue par l'un ou l'autre équipement esclave annulera tous les datagrammes établis dans *les deux* équipements esclaves du module CMM. Il n'y a pas d'interaction entre les datagrammes établis sur les deux ports d'un module CMM711 (Série 90–70). Il n'y a pas d'interaction non plus entre les datagrammes établis dans un module CMM et l'équipement esclave interne de la CPU, ou entre deux modules CMM distincts.

4. PROTOCOLE SNP–X

Une description complète du protocole SNP–X est fournie ci–après pour les utilisateurs qui développent des gestionnaires de communication SNP–X. Ce paragraphe traite des sujets suivants :

- Présentation du protocole SNP–X
- Séquence du protocole SNP–X
- Structure des messages SNP–X
- Commandes SNP–X
- Table des états SNP–X esclave

4.1. PRÉSENTATION DU PROTOCOLE SNP–X

Le protocole SNP–X est un protocole simple qui permet de transférer rapidement des données entre des API GE Fanuc et divers équipements. SNP–X fournit un transfert de données significativement plus rapide que le protocole SNP. Il ne bénéficie toutefois pas du jeu de services étendu offert sous SNP.

SNP–X est un protocole maître–esclave half–duplex qui utilise les interfaces électriques RS–485 et RS–232. Les équipements SNP–X peuvent être configurés en mode point–à–point (un seul équipement maître et un seul équipement esclave) ou multipoint (un seul équipement maître et plusieurs équipements esclaves). Un seul maître peut figurer dans une configuration multipoint. Lorsque plusieurs équipements esclaves se partagent une liaison multipoint, chacun d'eux est identifié par un ID SNP unique.

Le port série interne de la CPU des API Série 90–20 et Série 90–30 ne peuvent être configurés que comme des équipements SNP et SNP–X esclaves utilisant l'interface RS–485. Tout port série d'un module CMM Série 90–30 ou Série 90–70 configuré pour fonctionner sous le protocole SNP supporte à la fois les protocoles SNP et SNP–X ; chaque port série peut être configuré pour fonctionner comme un équipement maître ou esclave utilisant l'interface RS–485 ou RS–232.

4.1.1. Session de communication SNP–X

Le protocole SNP–X exige l'établissement d'une session de communication, appelée session SNP–X. Pour établir une session SNP–X, l'équipement maître commence par envoyer un Break long. Une fois le délai spécifié par le temporisateur T4 écoulé, l'équipement maître envoie une demande de connexion X à l'équipement esclave. L'équipement spécifié retourne une réponse de connexion X à l'équipement maître pour compléter l'établissement de la session SNP–X. Une session SNP–X signifie que le protocole SNP–X est maintenant en vigueur et que des données peuvent être transférées via des commandes SNP–X. La session SNP–X reste active jusqu'au prochain Break long.

Une session SNP–X peut être établie entre l'équipement maître et tout équipement esclave. A la différence des sessions SNP, des sessions SNP–X peuvent également être établies avec tous les équipements SNP–X esclaves sur une liaison série multipoint ; les commandes SNP–X utilisent ensuite l'ID SNP qui identifie l'équipement esclave désiré.

Le même réseau peut fonctionner sous protocole SNP ou SNP–X, suivant le type de session de communication spécifié (SNP ou SNP–X). Une seule session (SNP ou SNP–X) peut être active à la fois. L'un ou l'autre type de session peut être établi via un Break long, suivi de la commande de connexion ou connexion X appropriée. Pendant une session SNP–X, seules les commandes SNP–X sont reconnues ; les commandes SNP sont ignorées. De la même façon, les commandes SNP–X sont ignorées pendant une session SNP.

4.1.2. Gestion des erreurs

SNP–X génère deux types d'erreurs : *les erreurs matérielles* et *les erreurs logicielles*.

Les erreurs matérielles sont bloquantes. Elles surviennent lorsque la session de communication ne peut pas continuer. Les erreurs matérielles englobent les erreurs de transmission série (erreurs de totaux de contrôle, de perte d'encadrement de trame, de parité ou BCC), ainsi que les erreurs de protocole (type de message, type ou longueur du message suivant). Lorsqu'un message de demande SNP–X ne peut pas être reçu correctement (exemple, une erreur de transmission série s'est produite), l'équipement esclave ne renvoie pas de réponse à l'équipement maître. Le maître attend que la temporisation de réponse se soit écoulée avant d'annuler la session de communication SNP–X. Si le message de demande SNP–X est reçu correctement mais que l'équipement esclave ne peut pas la satisfaire, il retourne un code d'erreur dans un message de réponse SNP–X. (Voir liste des codes d'erreurs mineures correspondant à l'erreur majeure 15 ; codes d'erreurs mineures 32–35). L'équipement maître, comme l'équipement esclave, annule la session de communication SNP–X lorsqu'une erreur matérielle est détectée ; les opérations ne peuvent reprendre qu'après établissement d'une nouvelle session de communication SNP–X.

Les erreurs logicielles sont des erreurs non bloquantes. L'équipement esclave renvoie un message de réponse SNP–X renfermant un code d'erreur à l'équipement maître. La session de communication SNP–X reste établie ; d'autres commandes SNP–X peuvent être générées. Les erreurs logicielles englobent les codes de demande incorrects, les demandes de service erronées, ainsi que les types de mémoire ou longueurs invalides.

4.1.3. Fonction de diffusion

Les messages de demande de connexion et de demande SNP–X renferment normalement l'ID SNP de l'équipement esclave auquel le message est destiné. Le protocole SNP–X fournit en outre un mécanisme pour diffuser un message de connexion ou de demande SNP–X auprès de tous les équipements esclaves sur une liaison série multipoint. Pour diffuser une demande SNP–X, un ID SNP spécial **(FFFFFFFFFFFFFFFF)** est spécifié dans le message. Les demandes qui sont diffusées ne reçoivent pas de réponse. C'est la raison pour laquelle seules les demandes de connexion X ou d'écriture X sont significatives dans ce contexte. Une demande de connexion X diffusée établit une session de communication SNP–X avec tous les équipements SNP–X esclaves sur une liaison série multipoint. (Les demandes SNP–X qui suivent peuvent ensuite être dirigées sur l'équipement désiré). Une demande Ecrire X diffusée envoie les mêmes données à tous les équipements SNP–X esclaves qui ont établi une session SNP–X.

4.1.4. Support des modems

Tous les équipements SNP–X Série 90 permettent que le signal RTS du port série soit utilisé comme signal de numérotation d'un modem. Le signal RTS est toujours excité avant chaque transmission de données pendant le délai de retournement du modem spécifié et supprimé après la transmission. Si le délai de retournement du modem n'a pas été défini à la valeur nulle, le signal RTS précèdera les données sortantes. Les ports série des modules CMM peuvent être configurés avec un contrôle de flux matériel (utilisation des signaux RTS/CTS), ou sans aucun contrôle de flux (le signal CTS est ignoré). Les équipements esclaves des CPU Série 90 ne supportent pas le contrôle de flux matériel ; le signal CTS est toujours ignoré.

4.1.5. Mot d'état de l'API esclave (état complémentaire)

Chaque réponse SNP–X positive renferme un mot qui indique le dernier mot d'état de l'API de l'équipement esclave. Ce mot est identique au mot d'état de l'API fourni dans les données d'état complémentaires de l'API Série 90. (Voir Tableau Données d'état complémentaires dans le Chapitre 6). Ce mot renferme entre autres des bits qui indiquent l'état de l'API (mode Run/Stop/désactivé), les conditions de dépassement du temps de cycle, et la présence de défauts. Le mot d'état de l'API esclave fourni dans la réponse SNP–X n'est valide que dans les réponses positives ; il est automatiquement mis à 0 dans les réponses d'erreur.

4.1.6. Bits d'accès à la mémoire de l'équipement esclave (bits d'état X)

L'interface utilisateur de l'équipement SNP–X esclave définit trois nouveaux bits d'état contigus, appelés *Bits d'état X*, dans la mémoire de l'API de l'équipement esclave. Ces bits sont mis à jour par l'équipement SNP–X esclave pour fournir toutes les informations relatives à la communication SNP–X à l'API esclave. Les bits d'état X sont définis comme suit :

(Bit n+2)	(Bit n+1)	(Bit n)
Ecrire X réussi	Lire X réussi	Session active

Le bit Session active indique qu'une communication SNP–X a été établie avec cet équipement esclave. Il reste à 1 tant que la session SNP–X est active. Le bit Lire X réussi ou Ecrire X réussi est mis à 1 pendant un cycle lorsque les commandes Lire X ou Ecrire X concernant cet équipement esclave ont abouti, respectivement.

L'adresse et le fonctionnement des bits d'état X diffèrent suivant que l'équipement esclave est dans le module CPU d'un API Série 90 ou dans un module CMM. Dans un module CPU Série 90, les bits d'état X résident toujours dans les adresses suivantes :

Adresse	%S19	%S18	%S17
Bit d'état X	Ecrire X réussi	Lire X réussi	Session active
Mnémonique	SNPX_WT	SNPX_RD	SNPXACT

Dans un module CPU Série 90, les bits d'état X sont toujours mis à jour par l'équipement esclave. Le bit Session active indique qu'une communication SNP–X a été établie avec cet équipement esclave. Il reste à 1 tant que la session SNP–X est active. Le bit Lire X réussi ou Ecrire X réussi est mis à 1 pendant un cycle lorsque les commandes Lire X ou Ecrire X ont abouti, respectivement.

Dans un module CMM Série 90, les bits d'état X résident à l'adresse spécifiée par l'utilisateur dans une table de références orientée Bit de la mémoire de l'API local. Un jeu de bits d'état X distinct doit être défini pour chaque port série du module CMM configuré comme équipement SNP–X esclave. Les bits d'état X ne sont accessibles qu'une fois que l'adresse a été spécifiée par la commande COMREQ Programmer adresse des bits d'état X (07003). Le bit Session active est toujours mis à jour par l'équipement esclave et ne peut pas être réinitialisé par l'application en diagramme en échelle. Les bits Lire X et Ecrire X réussi ne peuvent être positionnés que par l'équiement esclave ; l'application en diagramme en échelle doit examiner et réinitialiser ces bits à chaque scrutation pour pouvoir détecter la prochaine indication fournie par les bits d'état X.

4.1.7. Temporisateurs

Le protocole SNP–X utilise les temporisateurs suivants. Vous remarquerez que le temporisateur T4 est utilisé exactement comme dans le protocole SNP.

4.1.7.1. T4

C'est l'intervalle de temps pendant lequel l'équipement SNP ou SNP–X maître doit attendre avant d'envoyer une commande de connexion ou de connexion X après avoir envoyé un Break long. Il permet ainsi à l'équipement esclave de configurer le port série et de se préparer à recevoir un message de connexion. Ce temporisateur n'est utilisé que par l'équipement maître. Sa valeur est généralement définie à 50 millisecondes. Si un transfert par modem est spécifié (en configurant le délai de retournement du modem à une valeur différente de zéro), la valeur du temporisateur T4 est de 600 millisecondes.

4.1.7.2. Délai de diffusion

C'est l'intervalle de temps minimal pendant lequel l'équipement SNP–X maître doit attendre après avoir envoyé un message de diffusion pour envoyer un autre message. Ce délai s'explique par le fait que sous le protocole SNP–X, les équipements esclaves ne répondent pas à un message de diffusion. Le délai de diffusion intervient typiquement après envoi d'une demande de connexion X diffusée. Il ne s'applique qu'à l'équipement SNP–X maître.

Pour le module CMM, la valeur du délai de diffusion est définie par défaut à la valeur **T2** indiquée par le paramètre de configuration SNP "Temps limit" (2 secondes par défaut). Cette valeur peut être modifiée par l'interface utilisateur de l'équipement SNP–X maître. La valeur optimale du délai de diffusion dépend des caractéristiques de scrutation de l'API esclave connecté le plus lent (vitesse de scrutation, fréquence et longueur de la fenêtre de communication, et charge système). Le délai de retournement du modem et le délai de transmission ne s'appliquent pas au délai de diffusion.

4.1.7.3. Temporisation de la réponse

C'est l'intervalle de temps maximal pendant lequel l'équipement SNP–X maître attendra le message retourné par l'équipement esclave. Si le message attendu n'est pas reçu dans cet intervalle de temps, l'équipement maître annulera la session de communication SNP–X ; une nouvelle session devra être établie pour toute autre communication SNP–X requise. Cette temporisation est utilisée dans l'attente d'une réponse SNP–X ou d'une réponse intermédiaire de la part de l'équipement esclave. (La temporisation de la réponse n'est pas utilisée dans les commandes de diffusion). Cette temporisation ne s'applique qu'à l'équipement SNP–X maître.

Pour le module CMM, la valeur du délai de diffusion est définie par défaut à la valeur **T2** indiquée par le paramètre de configuration SNP "Temps limit" (2 secondes par défaut). Cette valeur peut être modifiée par l'interface utilisateur de l'équipement SNP–X maître. La valeur optimale de la temporisation de la réponse dépend des caractéristiques de scrutation de l'API esclave (vitesse de scrutation, fréquence et longueur de la fenêtre de communication, charge système, et éventuelle détection et correction des défaillances de communication). Le temps requis pour transmettre le plus long message de réponse SNP–X (1015 octets) à la vitesse configurée est ajouté en interne à la valeur de temporisation de la réponse. Tout délai de retournement du modem ou délai de transmission non défini à zéro est également ajouté en interne à la valeur de temporisation de la réponse.

4.1.7.4. Temporisation du tampon

C'est l'intervalle de temps maximal pendant lequel l'équipement SNP–X esclave attendra le message SNP–X Tampon de l'équipement maître déporté. Si le message Tampon n'est pas reçu dans cet intervalle de temps, l'équipement esclave annulera la session de communication SNP–X ; une nouvelle session devra être établie pour toute autre communication SNP–X requise. Cette temporisation est utilisée chaque fois qu'un message Tampon est attendu, dans des demandes directes ou diffusées. La temporisation du tampon ne peut être ni configurée, ni spécifiée par l'interface utilisateur ; elle est calculée de façon interne pour être égale à 10 seconde plus le temps requis pour transmettre le plus long message SNP–X Tampon (1008 octets) à la vitesse configurée. Cette temporisation ne s'applique qu'à l'équipement SNP–X maître.

Le délai de retournement du modem et le délai de transmission (que l'équipement SNP–X ajoute éventuellement à la temporisation de la réponse et au délai de diffusion) sont décrits ci–dessous :

4.1.7.5. Délai de retournement du modem

C'est l'intervalle de temps requis par les modems intermédiaires pour inverser la liaison série. Ce délai prend la valeur par défaut configurée pour les modules CMM ou CPU Série 90. (Pour les modules CMM, cette valeur est spécifiée par le paramètre de configuration SNP "Temps limit"). Pour les équipements SNP–X maîtres du module CMM, cette valeur peut être modifiée par l'interface utilisateur de l'équipement SNP–X maître.

La valeur du délai de retournement du modem est également utilisée pour spécifier l'intervalle de temps pendant lequel le signal RTS restera activé avant la transmission des données. Il constitue un "signal de numérotation" pour les modems et est utilisé par tous les équipements SNP–X maîtres et esclaves.

4.1.7.6. Délai de transmission

C'est l'intervalle de temps requis pour que les données soient transférées de l'équipement initiateur à l'équipement répondeur. (Ce temps est également appelé "temps de câble"). Il reflète les délais inhabituels dans la voie de transmission des données, lorsque les liaisons satellites sont utilisées pour transférer les données par exemple. Pour le module CMM, ce délai est défini à zéro par défaut. Pour un équipement maître, cette valeur par défaut peut être modifiée par l'interface utilisateur de l'équipement SNP–X maître.

4.1.8. Break long

L'équipement SNP–X maître envoie un Break long pour demander à tous les équipements esclaves sur la liaison série de se préparer à établir une nouvelle session de communication. Les équipements esclaves annulent alors toutes les sessions de communication actives et se préparent à recevoir un message de connexion ou de connexion X. L'équipement maître génère un Break long en maintenant la ligne de transmission à l'état "Espace" pendant un laps de temps équivalent au temps de transmission de trois caractères à la vitesse sélectionnée. Si une transmission par modem est signalée (en configurant le délai de retournement du modem à une valeur non nulle), la durée du Break long est de 400 millisecondes pour qu'il ait le temps d'être transmis par les modems. Le Break long est utilisé de la même manière par les protocoles SNP et SNP–X.

4.1.9. Code de contrôle de bloc (BCC)

Le Code de Contrôle de Bloc (BCC) est le dernier octet transmis dans un message SNP–X et il est utilisé pour vérifier l'intégrité du contenu du message. La valeur BCC est calculée sur tous les octets du message, excepté le dernier qui correspond au BCC lui–même. A réception d'un message SNP–X, la valeur BCC est automatiquement recalculée et comparée à celle transmise dans le message.

La valeur BCC est calculée en combinant par des portes OU exclusif (XOR) successives chaque octet suivant dans le message et en décalant la valeur BCC cumulée d'un bit vers la gauche ; cette procédure est répétée sur tous les octets du message jusqu'à ce qu'ils aient été tous vérifiés.

L'algorithme de calcul du BCC est le suivant :

Initialiser l'octet BCC Temp à zéro
Pour le premier au dernier octet du message à vérifier
    XOR l'octet BCC Temp avec le premier octet du message
    Décaler l'octet BCC Temp d'un bit vers la gauche (avec bouclage)
    Passer à l'octet suivant dans le message
Finir pour
L'octet BCC Temp renferme maintenant la valeur BCC finale

Un exemple de calcul de l'octet BCC en langage "C" est fourni ci–dessous :

```
byte procedure calc_bcc( msg, len )
 byte * msg;    /* Pointeur sur début du message SNP–X  */
 int     len;    /* Nombre d'octets sur lequel le BCC doit être calculé */
{
        int  i;
        int  bcc; /* Octet BCC Temp; plus long qu'un octet pour */
                  /*   gérer le bouclage lors de la rotation  */
        i = 0;
        bcc = 0;

        while ( i < len )
        {
                  /* XOR l'octet BCC Temp avec l'octet du message  */
                  bcc = ( bcc ^ msg[i] );

                  /* Décaler l'octet BCC Temp d'un bit vers la gauche   */
                  bcc = ( bcc << 1 );
                  /* (Introduire octet PF dans octet Pf pour la rotation)  */
                  bcc = bcc + ( ( bcc >> 8 ) & 0x01 );

                  /* Incrémenter i pour passer à l'octet suivant dans le msg  */
                  i++;
        }

        /* Retourner l'octet BCC calculé  */
        return( (byte)bcc );
}
```

4.2. SÉQUENCE DU PROTOCOLE SNP–X

La séquence de messages utilisée par le protocole SNP–X pour établir une session de communication SNP–X et pour transférer des données est décrite ci–dessous. Tous les échanges de messages se résument à des demandes et à des réponses ; l'équipement maître envoie les demandes et l'équipement esclave retourne les réponses.

Les commandes SNP–X peuvent être *directes* ou *diffusées.* Les commandes SNP–X *directes* sont adressées à un équipement esclave spécifique et ne sont acceptées que par cet équipement. Les commandes directes consistent en une demande et une réponse. Une commande SNP–X directe renferme un seul message de demande et éventuellement, un message Tampon pour les données. La réponse ne renferme qu'un seul message de réponse.

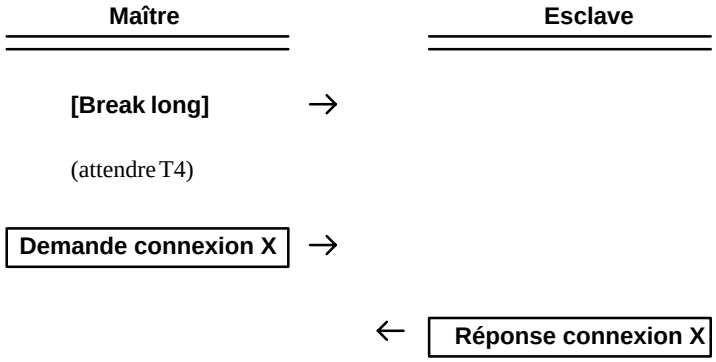
Les commandes SNP–X diffusées renferment une adresse de diffusion spéciale et sont acceptées par tous les équipements SNP–X esclaves sur une liaison série multipoint. Une commande SNP–X diffusée renferme un seul message de demande et éventuellement, un message Tampon pour les données. Pour éviter toute confusion au niveau de l'équipement maître, les équipements esclaves ne renvoient jamais de réponse à une commande diffusée. Seules certaines commandes SNP–X peuvent être diffusées. Etant donné que les commandes diffusées neutralisent les réponses, une commande SNP–X pour lire des données n'a pas de sens par exemple.

Les équipements maître et esclaves examinent chaque message reçu pour s'assurer qu'il ne renferme pas d'erreurs. Les erreurs de transmission série (erreurs de parité, de perte d'encadrement de trame/de séquence, BCC) sont bloquantes ; comme le message reçu n'était pas correct, aucune réponse ne peut être générée. Les violations du protocole (type de message erroné, informations sur le message suivant erronées) constituent également des erreurs bloquantes, bien que dans ce cas, une réponse d'erreur puisse être retournée. Toute erreur bloquante met immédiatement fin à la session SNP–X. Les erreurs de demande de service (code de demande, paramètres ou données invalides) ne sont pas bloquantes ; une réponse d'erreur est retournée et la session SNP–X reste active.

Si l'équipement esclave détecte une erreur et peut retourner une réponse (exemple, le message a été correctement reçu de l'équipement maître, quelque soit son contenu), il renvoie un code d'erreur dans le message Réponse X. Si une erreur identifiable est détectée dans un message de demande X qui utilise un tampon de données, un message de réponse X renfermant le code de l'erreur est retourné au lieu du message de réponse intermédiaire normal.

4.2.1. Etablissement d'une session

Pour établir une session de communication SNP–X, l'équipement maître envoie d'abord un Break long, attend l'intervalle de temps défini pour **T4**, et envoie un message de demande de connexion X. L'équipement esclave reconnaît toujours un Break long ; il réinitialise ses circuits de réception et se prépare à recevoir le message de demande de connexion X annoncé par le Break long. Si le message de demande de connexion X lui est destiné, l'équipement esclave renvoie une réponse de connexion X à l'équipement maître. Tous les autres équipements esclaves ignorent le message de connexion X. L'ID SNP Nul est accepté par tous les équipements esclaves, comme s'il s'agissait de leur propre ID SNP. L'utilisation de l'ID SNP nul (ID de diffusion) dans un message de connexion X est décrite dans "Commandes diffusées", ci–après. La séquence des messages du protocole SNP–X pour établir une session de communication SNP–X est illustrée ci–dessous :

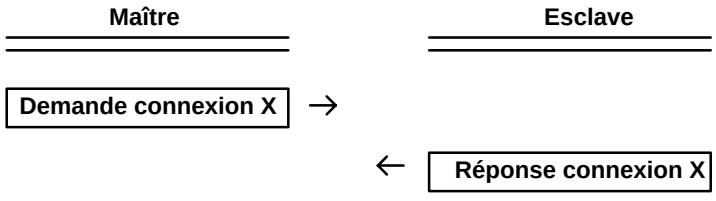


Une fois la session de communication SNP–X établie, les commandes SNP–X peuvent être utilisées pour transférer des donnés.

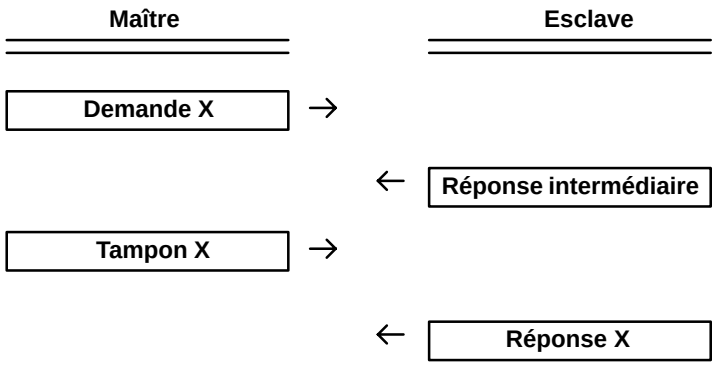
4.2.2. Commandes directes

Lorsque l'équipement maître envoie un message SNP–X contenant un ID SNP spécifique, seul l'équipement esclave identifié par cet ID SNP reçoit ce message ; il retourne alors le message de réponse approprié à l'équipement maître. Tous les autres équipements ignorent le message. L'ID SNP Nul est accepté par tous les équipements esclaves à la place de leur ID SNP respectif.

Toute commande Lire X, ou Ecrire X renfermant jusqu'à deux octets de données tient dans une seule demande SNP–X. L'équipement maître envoie un message Demande X et l'équipement esclave retourne un message Réponse X qui met fin à la commande. Aucun message Tampon X n'est utilisé. Toutes les données de la réponse tiennent dans un seul message Réponse X. La séquence des messages du protocole SNP–X pour une commande SNP–X sans tampon de données est illustrée ci–dessous :



Une commande Ecrire X avec plus de deux octets de données ne tient pas dans un seul message de demande SNP–X. L'équipement maître envoie un message de demande X et indique qu'un message Tampon suivra ; l'équipement esclave retourne un message de réponse spécial appelé Réponse intermédiaire. L'équipement maître envoie alors les données dans un message Tampon X et l'équipement esclave retourne le message de réponse X qui met fin à la demande. La séquence des messages du protocole SNP–X pour une commande SNP–X avec tampon de données est illustrée ci–dessous :



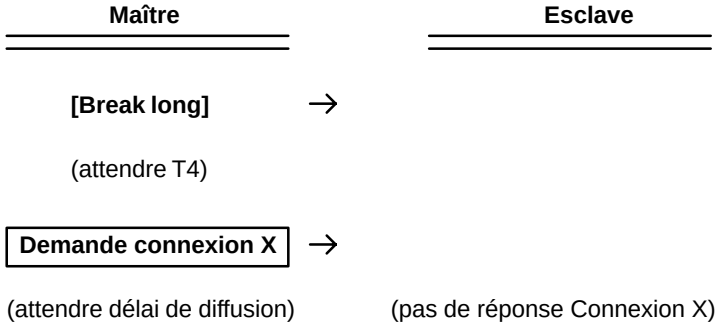
4.2.3. Commandes diffusées

Lorsque l'équipement maître envoie un message SNP–X (connexion X ou demande X) qui renferme l'ID SNP de diffusion spécial à la place d'un ID SNP particulier, tous les équipements esclaves sur la liaison série reçoivent le message. Si c'est un message de connexion X, une session de communication SNP–X sera établie avec chaque équipement esclave. Si c'est une demande X, tous les équipements esclaves qui ont précédemment établi une session SNP–X recevront et traiteront la commande. Les équipements SNP–X esclaves ne retournent aucun message en réponse à un message de diffusion. C'est pourquoi l'équipement maître ne peut pas détecter d'erreurs dans un équipement esclave à la suite d'une commande de diffusion.

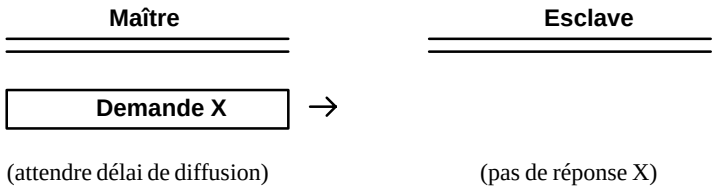
Cette absence de réponse des équipements esclaves à une commande de diffusion oblige l'équipement maître à attendre que le **Délai de diffusion** soit écoulé après chaque message envoyé pour laisser aux équipements esclaves le temps de la traiter.

Des commandes directes peuvent également être générées à la suite d'une demande de connexion X diffusée. Vous pouvez par exemple interroger rapidement tous les équipements esclaves à l'aide d'une commande de connexion X diffusée et établir ainsi des sessions de communication SNP–X simultanées avec tous les équipements esclaves sur la liaison série, puis adresser ensuite des commandes Lire X et Ecrire X à des équiquements esclaves individuels.

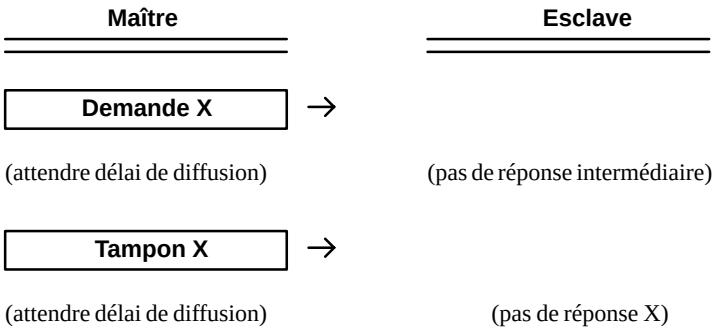
Pour les demandes de connexion X diffusées, la séquence du protocole SNP–X est la suivante :



Pour les commandes X diffusées sans tampon de données, la séquence du protocole SNP–X est la suivante :



Pour les commandes X diffusées avec tampon de données, la séquence du protocole SNP–X est la suivante :



4.3. STRUCTURE DES MESSAGES SNP–X

Les messages SNP–X requis pour exécuter chaque commande SNP–X sont décrits ci–dessous.

Une commande SNP–X est générée par un seul message de demande (Demande X). Si toutes les données de la commande ne peuvent pas être transmises dans le message de demande, les données sont envoyées dans un message Tampon (Tampon X). Un message de réponse (Réponse X), renfermant tous les états et les données de la réponse, met fin à la commande SNP–X. Le protocole SNP–X empêche les équipements esclaves de retourner une réponse à une commande de diffusion.

Chaque commande SNP–X respecte un format de protocole général. Tous les messages SNP–X se décomposent en trois parties :

- En–tête
- Données de la commande
- Fin de Trame

En–tête	Données de la commande	Fin de Trame

Le contenu des paragraphes qui suivent varie suivant le type de message : Message de demande SNP–X (Demande X), Message de réponse SNP–X (Réponse X), ou Tampon de données SNP–X (Tampon X).

4.3.1. Structure des messages de demande X

Le message de demande SNP–X (Demande X) est envoyé par l'équipement maître pour spécifier le service désiré et les paramètres. Il peut renfermer jusqu'à deux octets de données. Si la commande en exige davantage, toutes les données sont transmises dans un message à part, appelé Tampon X.

Tous les messages Demande X renferment exactement 24 octets qui se décomposent comme suit :

En–tête

Début de message	Type de message
Octet 1	Octet 2

Données de la commande

ID SNP	Code de la demande	Données spécifiques à la commande
Octets 3–10	Octet 11	Octets 12–18

Fin de Trame

Fin de bloc	Type du message suivant	Longueur du message suivant	Non utilisé	Code BCC
Octet 19	Octet 20	Octets 21, 22	Octet 23	Octet 24

Les contenus de l'en–tête, des données de la commande et de la fin de trame des messages Demande X sont décrits ci–dessous :

En–tête			
Octet 1	Début de message	1Bh	Caractère Début de message.
Octet 2	Type de message	58h	Message SNP–X (58h = 'X').
Données de la commande			
Octets 3–10	ID SNP		ID SNP de l'équipement esclave SNP–X désiré.
Octet 11	Code de la demande		Code de la demande de service désirée.
Octets 12–18	Données spécifiques à la commande		(Dépendent de la commande).
Fin de Trame			
Octet 19	Fin de bloc	17h	Caractère Fin de bloc.
Octet 20	Type du message suivant		Type du message suivant (0 ou 54h = 'T').
Octets 21,22	Longueur du message suivant		Longueur du message suivant (en octets).
Octet 23	Non utilisé	0	Non utilisé (toujours à 0).
Octet 24	Code BCC		Code BCC calculé pour ce message.

Le champ "ID SNP" renferme l'ID SNP de l'équipement esclave auquel ce message est envoyé. Ce champ peut également renfermer l'ID SNP Nul (les 8 octets à 0h) ou l'ID SNP de diffusion (les 8 octets à **FF**h).

Les champs "Code de la demande" et "Données spécifiques à la commande" varient avec chaque commande SNP–X.

La fin de trame du message de la demande X indique si un message Tampon de données sera utilisé. Si oui, le champ "Type du message suivant" est défini à **54**h, type du message Tampon X ; le champ "Longueur du message suivant" indique la longueur du message Tampon X complet (en–tête, donnée et fin de trame). Si non, ces champs sont définis à zéro dans la fin de trame de la demande X.

4.3.2. Structure des messages Réponse X

L'équipement SNP–X esclave transmet un message Réponse SNP–X (Réponse X) après avoir traité un message Demande X qui lui était adressé pour mettre fin à la Demande X, ou pour permettre à l'équipement maître d'envoyer d'autres données. Le message Réponse X renferme le mot d'état de l'API esclave (décrit précédemment), un code d'erreur (octets des codes d'erreurs majeures et mineures), et jusqu'à 1000 (décimal) octets de données.

Le message Réponse X spécial, appelé Réponse intermédiaire et spécifié par une valeur Type de message spéciale, est utilisé pour accuser réception d'un message Demande X qui sera suivi d'un message Tampon de données. La réponse intermédiaire ne renferme aucune donnée ; les champs Mot d'état de l'API et Code d'erreur ne signifient rien. Le seul rôle du message de réponse intermédiaire est d'accuser réception du message Demande X et de permettre la transmission du message Tampon de données.

Si l'équipement esclave a détecté une erreur, elle est indiquée par un code d'erreur différent de zéro. Le mot d'état de l'API est défini à zéro et le message ne renferme aucune donnée (Longueur des données définie à zéro).

Tous les messages Réponse X se décomposent comme suit :

En–tête

Début de message	Type de message
Octet 1	Octet 2

Données de la commande

Code de la réponse	Mot d'état de l'API	Code d'erreur	Longueur des données	Données de réponse facultatives
Octets 3	Octets 4, 5	Octets 6, 7	Octets 8, 9	Octets 10 – N

Fin de Trame

Fin de bloc	Type du message suivant	Longueur du message suivant	Non utilisé	Code BCC
Octet N+1	Octet N+2	Octets N+3, N+4	Octet N+5	Octet N+6

Les contenus de l'en–tête, des données de la commande et de la fin de trame des messages Réponse X sont décrits ci–dessous :

En–tête			
Octet 1	Début de message	1Bh	Caractère Début de message.
Octet 2	Type de message		Message SNP–X (58h = 'X', ou 78h = 'x').
Données de la commande			
Octet 3	Code de la réponse		Code de la réponse (= Code de la demande + 80h).
Octets 4,5	Mot d'état de l'API esclave		Mot d'état de l'API fourni par l'équipement esclave.
Octets 6,7	Code d'erreur		Code d'erreur (00 00 = succès).
Octets 8,9	Longueur des données		Longueur des données en octets (0–1000).
Octets 10–N	Données facultatives		Données facultatives (0–1000 octets).
Fin de Trame			
Octet N+1	Fin de bloc	17h	Caractère Fin de bloc.
Octet N+2	Type du message suivant	0	Type du message suivant (toujours à 0).
Octets N+3,4	Longueur du message suivant	0	Longueur du message suivant (toujours à 0).
Octet N+5	Non utilisé	0	Non utilisé (toujours à 0).
Octet N+6	Code BCC		Code BCC calculé pour ce message.

Le champ "Type de message" est défini à **58**h pour un message Réponse X ou à **78**h pour un message Réponse intermédiaire.

Le champ "Code de la réponse" indique le code de la demande X qui a généré cette commande. La valeur du code de la réponse est définie à la valeur du code de la demande, avec l'octet de poids fort toujours à 1.

Lorsque le message Réponse X a abouti, le champ "Mot d'état de l'API" renferme le mot d'état de l'API fourni par l'équipement esclave. Pour une Réponse X d'erreur, ce champ est à 0. Pour plus d'informations sur le contenu du mot d'état de l'API esclave, voir Tableau 6.13.

Le champ "Code d'erreur" renferme deux octets. Le premier octet (octet 6) renferme le code de l'erreur majeure (00 indique un succès) ; le second octet (octet 7) renferme le code de l'erreur mineure. La signification du code de l'erreur mineure dépend de la valeur du code de l'erreur majeure.

Le champ "Longueur des données" est toujours présent ; ce champ renferme le nombre d'octets de données qui suit immédiatement dans le message. Ce champ est à 0 lorsque le message ne renferme aucune donnée.

Le champ "Données facultatives" peut ne pas être présent. La taille de ce champ est indiquée par le champ "Longueur des données". Si le champ "Longueur des données" est à 0, le champ "Données facultatives" n'est pas présent.

4.3.3. Structure du message Tampon X

Le message Tampon de données SNP–X (Tampon X) est envoyé par l'équipement maître pour fournir des données supplémentaires relatives à la commande spécifiée dans le message Demande X précédent. Le message Tampon X n'est envoyé qu'après qu'un message de réponse intermédiaire ait été reçu de l'équipement esclave. Les messages Tampon X peuvent renfermer de 1 à 1000 (décimal) octets de données, et se décomposent comme suit :

En–tête

Début de message	Type de message
Octet 1	Octet 2

Données de la commande

Données facultatives
Octets 3 – N

Fin de Trame

Fin de bloc	Type du message suivant	Longueur du message suivant	Non utilisé	Code BCC
Octet N+1	Octet N+2	Octets N+3, N+4	Octet N+5	Octet N+6

Les contenus de l'en–tête, des données de la commande et de la fin de trame des messages Tampon X sont décrits ci–dessous :

En–tête			
Octet 1	Début de message	1Bh	Caractère Début de message.
Octet 2	Type de message	54h	Message SNP–X (54h = 'T').
Données de la commande			
Octets 3–N	Données facultatives		Données facultatives (1–1000 octets).
Fin de Trame			
Octet N+1	Fin de bloc	17h	Caractère Fin de bloc.
Octet N+2	Type du message suivant	0	Type du message suivant (toujours à 0).
Octets N+3,4	Longueur du message suivant	0	Longueur du message suivant (toujours à 0).
Octet N+5	Non utilisé	0	Non utilisé (toujours à 0).
Octet N+6	Code BCC		Code BCC calculé pour ce message.

La taille du champ "Données facultatives" dépend du volume de données dans le message. La taille du message Tampon X complet a été spécifiée par le champ "Longueur du message suivant" dans le message Demande X précédent.

4.4. COMMANDES SNP–X

Les messages échangés sous protocole SNP–X pour chaque commande SNP–X sont décrits ci–après. Les commandes directes et diffusées sont expliquées. Des exemples avec des valeurs réelles sont également fournis pour chaque commande.

4.4.1. Commande Connexion X

Avant que des données puissent être transférées entre des équipements SNP–X, une session de communication SNP–X doit être établie par l'équipement maître avec l'équipement esclave. L'équipement maître envoie un Break long et attend que l'intervalle de temps défini pour T4 se soit écoulé avant d'envoyer une demande de connexion X pour établir une session SNP–X avec l'équipement esclave.

La commande de connexion X peut être adressée à un équipement esclave particulier, ou diffusée à tous les équipements esclaves sur la liaison série. Pour l'adresser à un équipement esclave particulier, spécifiez l'ID SNP de l'équipement désiré. (Dans une configuration point–à–point, l'ID SNP Nul peut être spécifié au lieu de l'ID SNP réel de l'équipement esclave ; il répondra à la demande comme si son propre ID SNP avait été spécifié). Lorsqu'elle est adressée à un équipement esclave particulier, toute demande de connexion X appelle une réponse de connexion X ; une session de communication SNP–X est ensuite établie avec l'équipement esclave. La réponse de connexion X ne renferme pas de code d'erreur ; en cas d'erreurs, la réponse de connexion X n'est pas retournée.

Pour diffuser une commande de connexion X à tous les équipements esclaves sur la liaison série, spécifiez l'ID SNP de diffusion. Les équipements esclaves ne répondent pas à une demande de connexion X diffusée. L'équipement maître ne peut donc pas détecter l'éventuel échec d'une commande de connexion X diffusée. Après avoir diffusé une demande de connexion X, l'équipement maître doit attendre que le délai de diffusion se soit écoulé avant d'envoyer la demande X suivante. A ce stade, la session de communication SNP–X est supposée être établie.

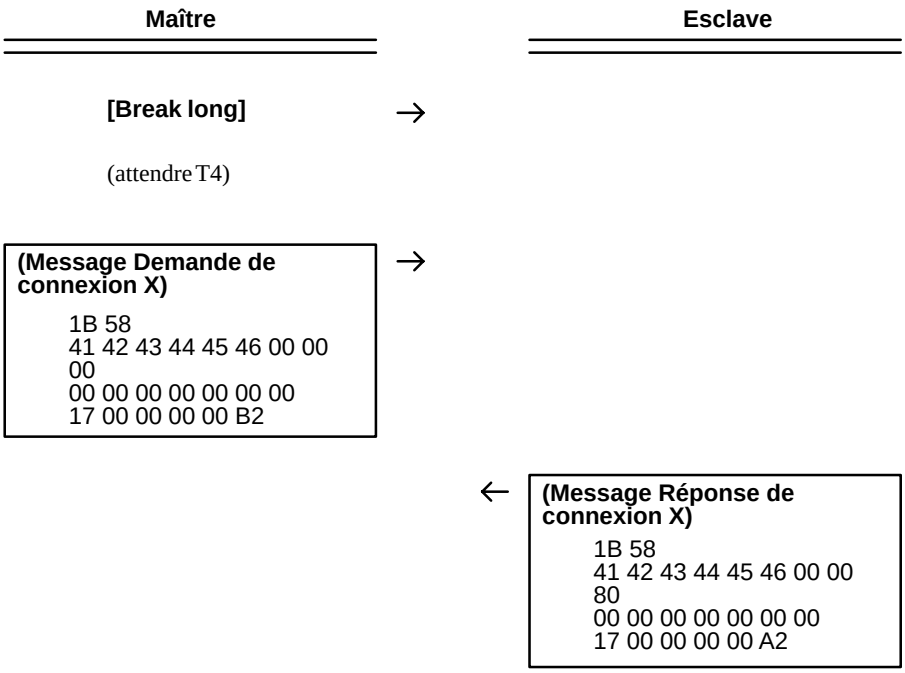
Une fois que la session de communication SNP–X a été établie, l'équipement esclave est prêt à accepter des commandes SNP–X. Vous pouvez générer autant de commandes ou de combinaisons de commandes SNP–X que désiré. La session SNP–X reste établie dans l'équipement esclave tant qu'aucune erreur bloquante ne survient, qu'un Break long n'est pas reçu ou que l'équipement n'est pas redémarré.

Les schémas et tableaux qui suivent décrivent les transactions impliquées dans l'établissement d'une session de communication SNP–X. Les deux variantes de la commande de connexion X (directe et diffusée) sont décrites. Les éléments–clé de l'établissement d'une session SNP–X sont la séquence Break long et la commande de connexion X.

Vous remarquerez que les messages Demande de connexion X et Réponse de connexion X ont la même structure (structure du message Demande X). Dans le message Réponse de connexion X, le code de la réponse est retourné dans le champ Code de la demande. Comme dans tous les messages de réponse, la valeur du code de la réponse de connexion X correspond au code de la demande (le code de la demande de connexion X est **0**h), avec l'octet de poids fort à 1. (C'est pourquoi le code de la réponse de connexion X est **80**h).

4.4.1.1. Exemple d'une commande de connexion X *directe*

Cet exemple montre les messages échangés lorsqu'une commande de connexion X directe est générée.

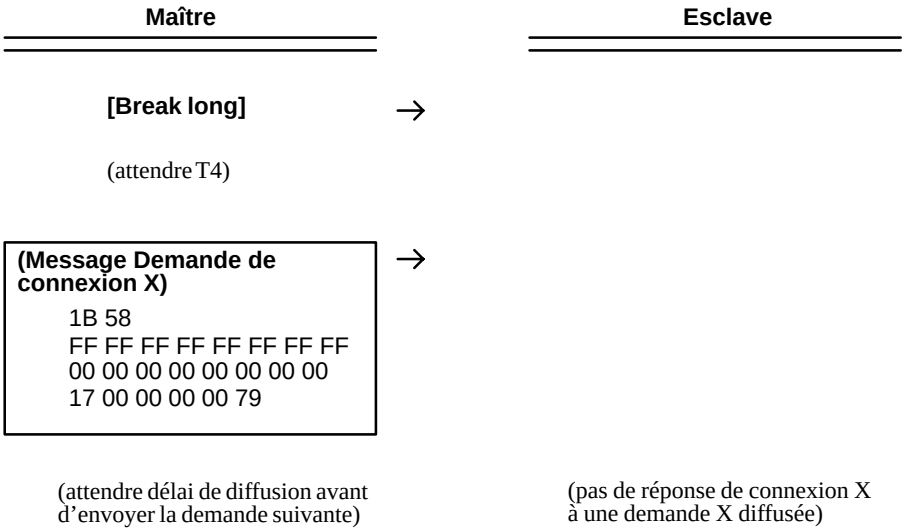


4.4.1.2. Explication de la commande de connexion X *directe*

Numéro de l'octet	Valeur hex	Description
Message Demande de connexion X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3–10	41 42 43 44 45 46 00 00	ID SNP de l'esclave destinataire.
11	00	Code de la demande de connexion X = 00h.
12–18	00 00 00 00 00 00 00	Non utilisés (toujours à 0).
19	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
20–23	00 00 00 00	Non utilisés (toujours à 0).
24	B2	Code BCC calculé pour cet exemple.
Message Réponse de connexion X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3–10	41 42 43 44 45 46 00 00	ID SNP de l'esclave destinataire.
11	80	Code de la réponse de connexion X = 80h.
12–18	00 00 00 00 00 00 00	Non utilisés (toujours à 0).
19	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
20–23	00 00 00 00	Non utilisés (toujours à 0).
24	A2	Code BCC calculé pour cet exemple.

4.4.1.3. Exemple d'une commande de connexion X *diffusée*

Cet exemple montre les messages échangés lorsqu'une commande de connexion X diffusée est générée. Vous remarquerez qu'il n'y a pas de message de réponse à une demande diffusée.


Maître
Esclave
[Break long]
(attendre T4)
(Message Demande de connexion X)
1B 58
FF FF FF FF FF FF FF FF
00 00 00 00 00 00 00 00
17 00 00 00 00 79
(attendre délai de diffusion avant d'envoyer la demande suivante)
(pas de réponse de connexion X à une demande X diffusée)

4.4.1.4. Explication de la commande de connexion X *diffusée*

Numéro de l'octet	Valeur hex	Description
Message Demande de connexion X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3–10	FF FF FF FF FF FF FF FF	ID SNP de diffusion.
11	00	Code de la demande de connexion X = 00h.
12–18	00 00 00 00 00 00 00	Non utilisés (toujours à 0).
19	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
20–23	00 00 00 00	Non utilisés (toujours à 0).
24	79	Code BCC calculé pour cet exemple.

4.4.2. Commande Lire X

La commande Lire X permet à l'équipement SNP–X maître de lire une zone continue dans une table de références définie dans la mémoire de l'API esclave. La demande spécifie le sélecteur de segments, le décalage et la longueur des données. Chaque commande Lire X génère un seul message de demande X et retourne un seul message de réponse X. La taille de la réponse X dépend du volume de données à lire. Le volume de données maximal qui peut être lu avec la commande Lire X est de 1000 (décimal) octets. Si l'équipement esclave détecte une erreur lors du traitement de la demande X, il retourne un message de réponse X renfermant le code d'erreur approprié.

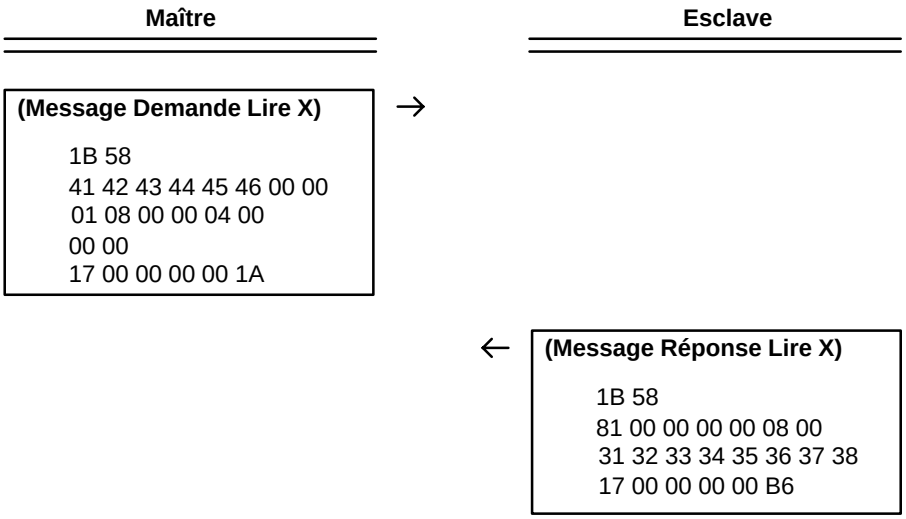
Les octets de données sont toujours alignés sur les octets de mémoire. Pour lire des données binaires individuelles, les adresses binaires correspondantes dans les octets de données doivent être fournies. %I4 par exemple correspond au quatrième bit de l'octet de données : 0000 1000 = 08h.

La commande Lire X ne peut être adressée qu'à un équipement esclave particulier en utilisant l'ID SNP de l'équipement désiré. (Dans une configuration point–à–point, l'ID SNP Nul peut être spécifié au lieu de l'ID SNP réel de l'équipement esclave ; il répondra à la demande comme si son propre ID SNP avait été spécifié). La commande Lire X ne peut pas être diffusée, étant donné que les équipements esclaves ne répondent à aucune commande de diffusion.

Le schéma et le tableau ci–après décrivent les transactions impliquées dans une commande Lire X.

4.4.2.1. Exemple d'une commande Lire X *directe*

Cet exemple montre les messages échangés lorsqu'une commande Lire X est générée.



4.4.2.2. Explication de la commande Lire X *directe*

Numéro de l'octet	Valeur hex	Description
Message Demande Lire X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3–10	41 42 43 44 45 46 00 00	ID SNP de l'esclave destinataire.
11	01	Code de la demande Lire X = 01h.
12	08	Sélecteur de segments : %R en mode Mot.
13–14	00 00	Décalage des données : 0000h = 0 (à partir de zéro) = %R1.
15–16	04 00	Longueur des données : 0004h = 4 registres.
17–18	00 00	Non utilisés (toujours à 0).
19	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
20–23	00 00 00 00	Non utilisés (toujours à 0).
24	1A	Code BCC calculé pour cet exemple.
Message Réponse Lire X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3	81	Code de la réponse Lire X = 81h.
4–5	00 00	Mot d'état de l'API. (00 00 pour cet exemple seulement)
6–7	00 00	Codes d'erreur (erreurs majeure et mineure).
8–9	08 00	Longueur des données dans la réponse (8 octets).
10–17	31 32 33 34 35 36 37 38	Données (%R1–%R4).
18	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
19–22	00 00 00 00	Non utilisé (toujours à 0).
23	B6	Code BCC calculé pour cet exemple.

4.4.3. Commande Ecrire X

La commande Ecrire X permet à l'équipement SNP–X maître d'écrire dans une zone continue d'une table de références définie dans la mémoire de l'API esclave. La demande spécifie le sélecteur de segments, le décalage et la longueur des données. Si deux octets de données ou moins doivent être écrits, un seul message Demande Ecrire X est transmis ; jusqu'à deux octets de données peuvent être chargés dans le message. Au–delà de deux octets, l'équipement SNP–X maître doit envoyer au message de demande X suivi d'un message Tampon. Dans ce cas, toutes les données sont chargées dans le message Tampon X ; le champ de données du message de demande n'est pas utilisé. A réception d'une demande X qui utilise un tampon de données, l'équipement esclave retourne un message de réponse intermédiaire à l'équipement maître ; ce dernier envoie alors le message Tampon X. Le volume de données maximal qui peut être chargé dans le message Tampon X est de 1000 (décimal) octets. L'équipement esclave met fin à la commande en renvoyant un message Réponse X à l'équipement maître. S'il détecte une erreur lors du traitement d'un message Demande X ou Tampon X, il renvoie un message Réponse X d'erreur avec le code d'erreur approprié.

Les octets de données sont toujours alignés sur les octets de mémoire. Pour écrire des données binaires individuelles, les adresses binaires correspondantes dans les octets de données doivent être fournies. %I4 par exemple correspond au quatrième bit de l'octet de données : 0000 1000 = 08h.

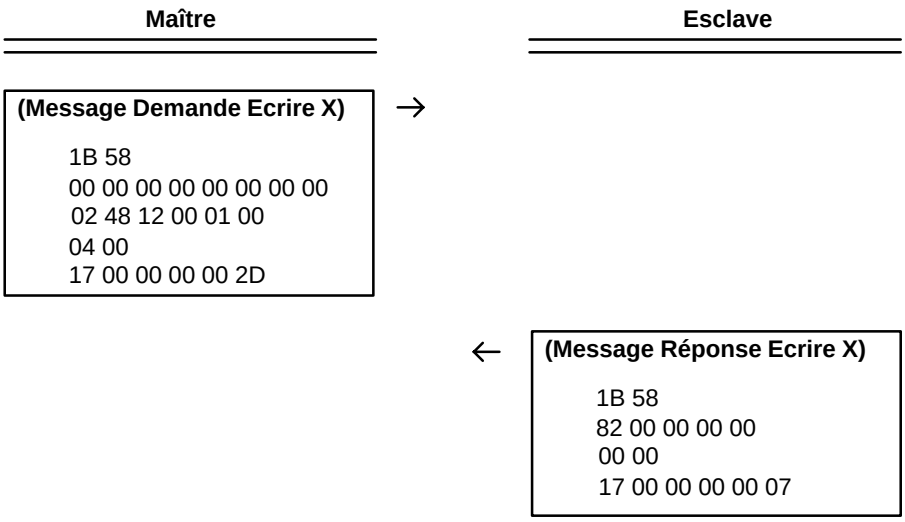
La commande Ecrire X peut être adressée à un équipement esclave particulier, ou diffusée à tous les équipements esclaves sur la liaison série. Pour l'adresser à un équipement esclave particulier, spécifiez l'ID SNP de l'équipement désiré. (Dans une configuration point–à–point, l'ID SNP Nul peut être spécifié au lieu de l'ID SNP réel de l'équipement esclave ; il répondra à la demande comme si son propre ID SNP avait été spécifié).

Pour diffuser une commande Ecrire X à tous les équipements esclaves sur la liaison série, spécifiez l'ID SNP de diffusion. Les équipements esclaves ne répondent pas à une demande d'écriture X diffusée. L'équipement maître ne peut donc pas détecter l'éventuel échec d'une commande Lire X diffusée. Après avoir diffusé une demande d'écriture X ou le message Tampon X qui lui est associé, l'équipement maître doit attendre que le délai de diffusion se soit écoulé avant d'envoyer le message suivant.

Les schémas et tableaux qui suivent décrivent les transactions impliquées dans une demande d'écriture X. Les deux variantes de la commande Ecrire X (directe et diffusée) sont décrites. Les exemples fournis expliquent ce qui se passe lorsque la commande Ecrire X est accompagnée d'un message Tampon X et lorsque le message Tampon X (optionnel) n'est pas utilisé.

4.4.3.1. Exemple d'une commande Ecrire X *directe* (2 octets ou moins)

Cet exemple montre les messages échangés lorsqu'une commande Ecrire X est générée, sans tampon de données.

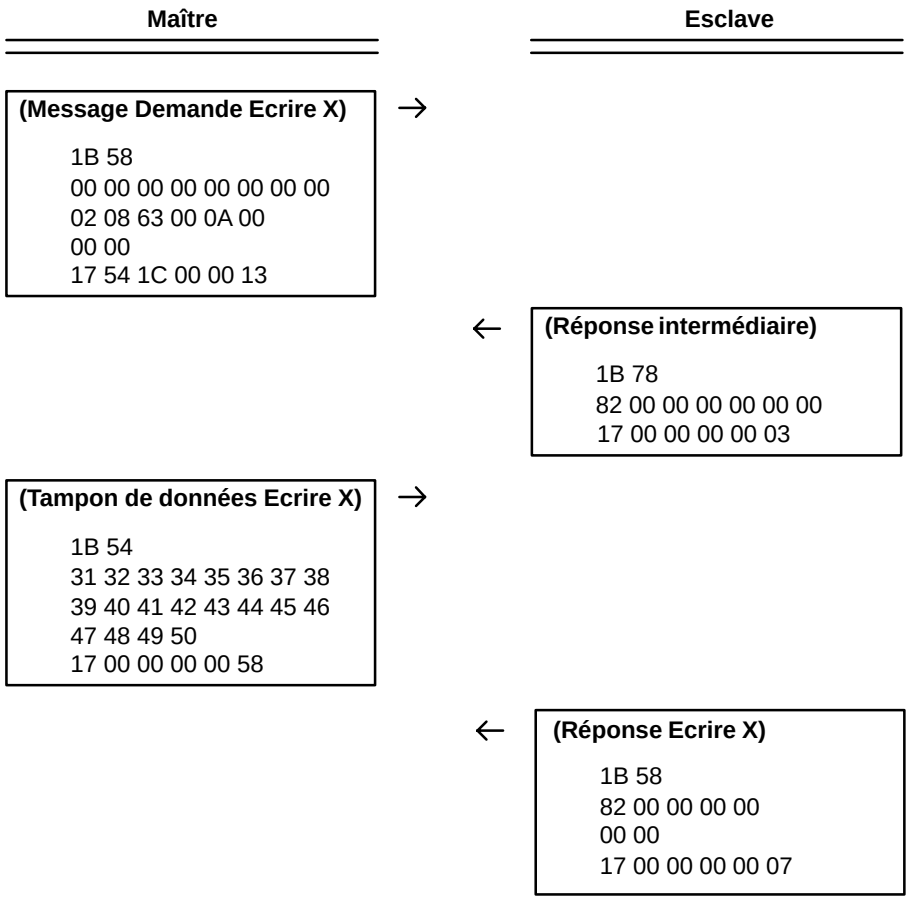


4.4.3.2. Explication de la commande Ecrire X *directe* (2 octets ou moins)

Numéro de l'octet	Valeur hex	Description
Message Demande Ecrire X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3–10	00 00 00 00 00 00 00 00	ID SNP de l'esclave destinataire.
11	02	Code de la demande Ecrire X = 02h.
12	48	Sélecteur de segments : %Q en mode Bit.
13–14	12 00	Décalage des données : 0012h = 18 (à partir de zéro) = %Q19.
15–16	01 00	Longueur des données : 0001h = 1 bit.
17–18	04 00	Données à écrire (1 mot max. = 2 octets = 16 bits).
19	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
20–23	00 00 00 00	Non utilisés (toujours à 0).
24	2D	Code BCC calculé pour cet exemple.
Message Réponse Ecrire X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3	82	Code de la réponse Ecrire X = 82h.
4–5	00 00	Mot d'état de l'API. (00 00 pour cet exemple seulement)
6–7	00 00	Code d'erreur (erreurs majeure et mineure).
8–9	00 00	Longueur des données (toujours à 0).
10	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
11–14	00 00 00 00	Non utilisés (toujours à 0).
15	07	Code BCC calculé pour cet exemple.

4.4.3.3. Exemple d'une commande Ecrire X *directe* (plus de 2 octets)

Cet exemple montre les messages échangés lorsqu'une commande Ecrire X directe comportant plus de deux octets de données est générée. La Demande X spécifie la transmission d'un message Tampon X à la suite, par l'équipement SNP–X maître. A réception de la demande X initiale, l'équipement esclave renvoie un message de réponse intermédiaire et attend le message Tampon X. A réception du message Tampon X, l'équipement SNP–X esclave écrit les données spécifiées dans la table de références de l'API. Il retourne ensuite le message Réponse X approprié pour mettre fin à la commande.



4.4.3.4. Explication de la commande Ecrire X *directe* (Plus de 2 octets)

Numéro de l'octet	Valeur hex	Description
Message Demande Ecrire X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3–10	00 00 00 00 00 00 00 00	ID SNP de l'esclave destinataire.
11	02	Code de la demande Ecrire X = 02h.
12	08	Sélecteur de segments : %R en mode Mot.
13–14	63 00	Décalage des données : 0063h = 99 (à partir de zéro) = %R100.
15–16	0A 00	Longueur des données : 000Ah = 10 mots.
17–18	00 00	Non utilisés (toujours à 0).
19	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
20	54	Type du message suivant = Tampon = 'T' = 54h.
21–22	1C 00	Longueur du message suivant (en octets) = 001Ch = 28.
23	00	Non utilisés (toujours à 0).
24	13	Code BCC calculé pour cet exemple.
Message Réponse intermédiaire		
1	1B	Caractère Début de message (1Bh).
2	78	Réponse intermédiaire SNP–X = 'x' = 78h.
3	82	Code de la réponse Ecrire X = 82h.
4–9	00 00 00 00 00 00	Non utilisés (toujours à 0).
10	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
11–14	00 00 00 00	Non utilisés (toujours à 0).
15	03	Code BCC calculé pour cet exemple.
Message Tampon de données X		
1	1B	Caractère Début de message (1Bh).
2	54	Commande SNP–X = 'T' = 54h.
3–22	31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50	20 octets de données à écrire dans %R100 – %R109.
23	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
24–27	00 00 00 00	Non utiliséss (toujours à 0).
28	58	Code BCC calculé pour cet exemple.
Message Réponse Ecrire X		
1	1B	Caractère Début de message (1Bh).
2	58	Réponse intermédiaire SNP–X = 'X' = 58h.
3	82	Code de la réponse Ecrire X = 82h.
4–5	00 00	Mot d'état de l'API. (00 00 pour cet exemple seulement)
6–7	00 00	Codes d'erreur (erreurs majeure et mineure).
8–9	00 00	Longueur des données (toujours à 0).
10	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
11–14	00 00 00 00	Non utilisés (toujours à 0).
15	07	Code BCC calculé pour cet exemple.

4.4.3.5. Exemple d'une commande Ecrire X *diffusée* pour 2 octets ou moins

Cet exemple montre les messages échangés lorsqu'une commande Ecrire X diffusée est générée sans tampon de données. Vous remarquerez qu'il n'y a pas de message Réponse à une demande diffusée.

Maître		Esclave
(Message Demande Ecrire X) 1B 58 FF FF FF FF FF FF FF FF 02 48 12 00 01 00 04 00 17 00 00 00 00 2D	→	
(Attendre délai de diffusion avant d'envoyer la demande suivante)		(Pas de Réponse Ecrire X à une commande Ecrire diffusée)

4.4.3.6. Explication de la commande Ecrire X *diffusée* (2 octets ou moins)

Numéro de l'octet	Valeur hex	Description
Message Demande Ecrire X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3–10	ff ff ff ff ff ff ff ff	ID SNP de diffusion.
11	02	Code de la demande Ecrire X = 02h.
12	48	Sélecteur de segments : %Q en mode Bit.
13–14	12 00	Décalage des données : 0012h = 18 (à partir de zéro) = %Q19.
15–16	01 00	Longueur des données : 0001h = 1 bit.
17–18	04 00	Données à écrire (1 mot max. = 2 octets = 16 bits).
19	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
20–23	00 00 00 00	Non utilisés (toujours à 0).
24	2D	Code BCC calculé pour cet exemple.

4.4.3.7. Exemple d'une commande Ecrire X *diffusée* (Plus de 2 octets)

Cet exemple montre les messages échangés lorsqu'une commande Ecrire X diffusée comportant plus de deux octets de données est générée. Cette commande exige un message de demande d'écriture suivi d'un message Tampon de données. Vous remarquerez qu'il n'y a ni réponse intermédiaire ni message de réponse final à une demande diffusée.

Maître	Esclave
[Message Demande Ecrire X] → 1B 58 FF FF FF FF FF FF FF FF 02 08 63 00 0A 00 00 00 17 54 1C 00 00 13	
(Attendre délai de diffusion avant avant d'envoyer le tampon de données)	(Pas de Réponse intermédiaire à une demande X diffusée)
[Tampon de données Ecrire X] → 1B 54 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 17 00 00 00 00 58	
(Attendre délai de diffusion avant d'envoyer la demande suivante)	(Pas de Réponse Ecrire X à une commande Ecrire diffusée)

4.4.3.8. Explication de la commande Ecrire X *diffusée* (Plus de 2 octets)

Numéro de l'octet	Valeur hex	Description
Message Demande Ecrire X		
1	1B	Caractère Début de message (1Bh).
2	58	Commande SNP–X = 'X' = 58h.
3–10	ff ff ff ff ff ff ff ff	ID SNP de diffusion.
11	02	Code de la demande Ecrire X = 02h.
12	08	Sélecteur de segments : %R en mode Mot.
13–14	63 00	Décalage des données : 0063h = 99 (à partir de zéro) = %R100.
15–16	0A 00	Longueur des données : 000Ah = 10 mots.
17–18	00 00	Non utilisés (toujours à 0).
19	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
20	54	Type du message suivant = Tampon = 'T' = 54h.
21–22	1C 00	Longueur du message suivant (en octets) = 001Ch = 28.
23	00	Non utilisé (toujours à 0).
24	13	Code BCC calculé pour cet exemple.
Message Tampon de données Ecrire X		
1	1B	Caractère Début de message (1Bh).
2	54	Commande SNP–X = 'T' = 54h.
3–22	31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50	20 octets de données à écrire dans %R100 – %R109.
23	17	Le caractère Fin de bloc (17h) marque le début de la fin de trame du message SNP–X.
24–27	00 00 00 00	Non utilisés (toujours à 0).
28	58	Code BCC calculé pour cet exemple.

4.5. TABLES DES ÉTATS SNP–X

Des tables d'états génériques correspondant au traitement du protocole dans les équipements SNP–X maître et esclaves sont décrites ci–après. Elles sont fournies pour aider ceux qui souhaitent développer leurs propres mises en oeuvre SNP–X maître ou esclave. Les tables d'états ne montrent que le traitement du protocole SNP–X ; elles ne suffisent pas à la mise en oeuvre complète des équipements SNP–X.

Des tables d'états distinctes sont fournies pour les équipements SNP–X maître et esclave. Chaque table indique les états de l'équipement et les événements d'entrée autorisées ; un sous–programme d'actions à entreprendre est indiqué pour chaque combinaison d'événement d'entrée et d'état courant. A la suite de chaque table, le pseudo–code du sous–programme d'actions utilisé pour cette table est fourni. Les noms des temporisateurs utilisés dans le pseudo–code correspondent à ceux décrits précédemment dans ce chapitre.

4.5.1. Table des états de l'équipement SNP–X maître

La table ci–dessous définit les actions entreprises par l'équipement SNP–X maître suivant les états :

ETAT EVENEMENT	Maître Inactif (1)	Attendre Rép Connexion X (2)	Maître Ouvrir (3)	Attendre Réponse Inter (4)	Attendre Rép X (5)
Cmd Utilisateur = Etablir nouvelle session SNP–X	Action A Etat (2)				
Réception Msg Rép Connexion X		Action B Etat (3/1)			
Cmd Utilisateur = Lire X ou Ecrire X			Action D Etat (4/5)		
Réception Msg Rép intermédiaire				Action E Etat (5/1)	
Réception Msg Réponse X				Action G Etat (3/1)	Action F Etat (3/1)
Expiration du délai de diffusion		Action C Etat (3)		Action H Etat (5)	Action C Etat (3)
Expiration Temporisation de la réponse		Action I Etat (1)		Action I Etat (1)	Action I Etat (1)

Pour toute intersection invalide dans cette table, tout traitement en cours dans l'équipement maître est annulé et l'équipement retourne à Etat (1).

4.5.2. Actions de l'équipement SNP–X maître

Les sous–programmes d'actions suivants sont utilisés par la table d'états de l'équipement SNP–X maître :

A Préparer et envoyer msg Connexion X à esclave
Si Demande Connexion X diffusée
Démarrer Temporisateur Délai de diffusion
Ou bien Demande Connexion X directe
Se préparer à lire Réponse Connexion X
Démarrer Temporisation de la réponse
Transition à Etat (2)

B Annuler Temporisation de la réponse
Si message Réponse Connexion X incorrect
Erreur! Annuler tout traitement en cours dans maître
Transition à Etat (1)
Ou bien (Cmd Utilisateur exécutée)
Transition à Etat (3) pour cmd suivante

C (Cmd Utilisateur exécutée ; toujours réussie)
Transition à Etat (3) pour cmd suivante

D Si Cmd Utilisateur = Ecrire X, aller chercher données à écrire
Préparer et envoyer msg Demande X à esclave
Si Demande X diffusée
Démarrer Temporisateur délai de diffusion
Ou bien Demande X directe
Se préparer à lire msg Réponse
Démarrer Temporisation de la réponse
Si Demande X avec tampon
Transition à Etat (4)
Ou bien Demande X sans tampon
Transition à Etat (5)

E Annuler Temporisation de la réponse
Si msg Réponse Intermédiaire incorrect
Erreur! Annuler tout traitement en cours dans maître
Transition à Etat (1)
Préparer et envoyer msg Tampon X à esclave
Se préparer à lire msg Réponse
Démarrer Temporisation de la réponse
Transition à Etat (5)

F Annuler Temporisation de la réponse
Si msg Réponse X incorrect
ou code erreur bloquante retourné
Erreur! Annuler tout traitement en cours dans maître
Transition à Etat (1)
Si erreur non bloquante
(le msg Réponse X renferme un code d'erreur)
Ou bien réponse positive
Si réponse Lire X
Distribuer données retournées par esclave
(Cmd Utilisateur exécutée)
Transition à Etat (3) pour cmd suivante

G Annuler Temporisation de la réponse
Si msg Réponse X incorrect
ou code d'erreur bloquante retourné
Erreur! Annuler tout traitement en cours dans maître
Transition à Etat (1)
Ou bien erreur non bloquante
(le msg Réponse X renferme un code d'erreur)
(Cmd Utilisateur exécutée)
Transition à Etat (3) pour cmd suivante

H Préparer et envoyer msg Tampon X à esclave
Démarrer Temporisateur délai de diffusion
Transition à Etat (5)

I Erreur de temporisation! Annuler tout traitement en cours dans maître
Transition à Etat (1)

4.5.3. Table des états de l'équipement SNP–X esclave

La table ci–dessous définit les actions entreprises par l'équipement SNP–X esclave suivant les états :

ETAT EVENEMENT	Attendre Break (1)	Attendre Connexion X (2)	Attendre Demande X (3)	Attendre Tampon X (4)
Réception Break long	Action A Etat (2)	Action A Etat (2)	Action A Etat (2)	Action A Etat (2)
Réception msg Connexion X		Action B Etat (3/1)		
Réception msg Demande X			Action C Etat (3/4/1)	
Réception msg Tampon X				Action D Etat (3/1)
Expiration Temporisation du tampon				Action E Etat (1)

Pour toute intersection invalide dans cette table, tout traitement en cours dans l'équipement esclave est annulé et l'équipement retourne à Etat (1)

4.5.4. Actions de l'équipement SNP–X esclave

Les sous–programmes d'actions suivants sont utilisés par la table d'états de l'équipement SNP–X esclave :

A Annuler tout traitement en cours dans esclave
Se préparer à recevoir un msg Connexion X
Transition à Etat (2)

B Si msg Connexion X incorrect
Erreur! Transition à Etat (1)
Si msg Connexion X destiné à un autre esclave
(Pas d'erreur) Transition à Etat (1)
Si ce n'est pas un msg Connexion X diffusé
Préparer et envoyer Rép Connexion X
Se préparer à recevoir msg Demande X suivant
Transition à Etat (3)

C Si erreur non bloquante dans msg Demande X
Préparer et envoyer msg Réponse X d'erreur
Si erreur bloquante
Annuler tout traitement en cours dans esclave
Transition à Etat (1)
Ou bien erreur non bloquante
Rester dans Etat (3)
Si msg Tampon X doit suivre
Si ce n'est pas une demande X diffusée
Préparer et envoyer msg Rép Intermédiaire
Démarrer Temporisation du tampon
Transition à Etat (4)
Exécuter service demandé (Lecture/Ecriture)
Si ce n'est pas une demande X diffusée
Préparer et envoyer msg Réponse X
Se préparer à recevoir msg Demande X suivant
Rester dans Etat (3)

D Annuler Temporisation du tampon
Si erreur non bloquante dans msg Tampon X
Préparer et envoyer msg Réponse X d'erreur
Si erreur bloquante
Annuler tout traitement en cours dans esclave
Transition à Etat (1)
Ou bien erreur non bloquante
Transition à Etat (3)
Exécuter service demandé (Lecture/Ecriture)
Si précédente Demande X n'était pas difffusée
Préparer et envoyer msg Réponse X
Se préparer à recevoir msg Demande X suivant
Transition à Etat (3)

E Erreur de temporisation du tampon! Annuler tout traitement en cours dans esclave
Transition à Etat (1)

Page laissée blanche intentionnellement

Chapitre 8 Schémas de câblage série

Ce chapitre fournit toutes les instructions nécessaires pour vous permettre de fabriquer des câbles communication série pour raccorder le module CMM à un autre équipement. Il traite des sujets suivants :

- 1 : Spécifications d’assemblage du câble
- 2 : Câbles RS–232
- 3 : Câbles RS–422
- 4 : Convertisseur/répéteur RS–422 isolé (IC655CCM590)

1. SPÉCIFICATIONS D'ASSEMBLAGE DU CÂBLE

La réalisation du raccordement du câble est à l'origine de la plupart des échecs de communication. Pour une performance optimale, fabriquez le câble à partir des éléments recommandés et en respectant les spécifications.

- Connecteur du câble pour ports 1 ou 2 du module CMM : connecteur mâle, type Sub–miniature D, Cannon DB25P (à souder) avec boîtier DB110963–3, ou connecteur RS–232 standard équivalent.
- Les connecteurs pour ports 1 et 2 se trouvent sur la carte CMM Série 90–70. Les deux ports sont fournis par un seul connecteur sur la carte CMM Série 90–30. Un câble en "Y" est fourni avec chaque carte CMM Série 90–30.
- Longueur (maximale) :
  - 15 mètres pour RS–232.
  - 15 mètres pour RS–422/RS–485 sans isolation sur l'extrémité déportée.
  - 1200 mètres pour RS–422/RS–485 avec isolation sur l'extrémité déportée.
- Ecran total.
- 0.2 mm^2 minimum (24 AWG).
- Connecteur vers équipement externe : spécifié par le fabricant de l'équipement externe.

Les câbles suivants fonctionnent de façon correcte pour des vitesses maximales de 19,2K Bds et des distances allant jusqu'à 1200 mètres pour RS–422/RS–485 :

Belden	9505
Belden	9184
Belden	9302
NEC	222P1SLCBT

Pour des distances inférieures à 15 mètres, presque tous les câbles à paire torsadée ou paire torsadée blindée fonctionneront, sous réserve que les paires câblées soient correctement connectées.

Avec l'interface RS–422/RS–485, les paires torsadées doivent être câblées de sorte que les deux signaux d'émission forment une paire torsadée et que les deux signaux de réception forment l'autre. Si ce principe n'est pas respecté, la diaphonie qui en résultera affectera la performance du système de communication.

Lorsque les câbles de communication passent par l'extérieur, des dispositifs de suppression de transitoires peuvent être utilisés pour limiter les risques d'endommagement dus à la foudre ou aux décharges d'électricité statique.

Attention

S'assurer impérativement que le module CMM et l'équipement auquel il est raccordé sont tous deux reliés à la même masse. Le non–respect de ce principe peut endommager l'équipement.

2. SCHÉMAS DE CÂBLAGE RS–232

Vous trouverez dans ce paragraphe le schéma des connexions RS–232 suivantes :

- Module CMM à module CMM
- Module CMM à Terminal Opérateur d'Interface

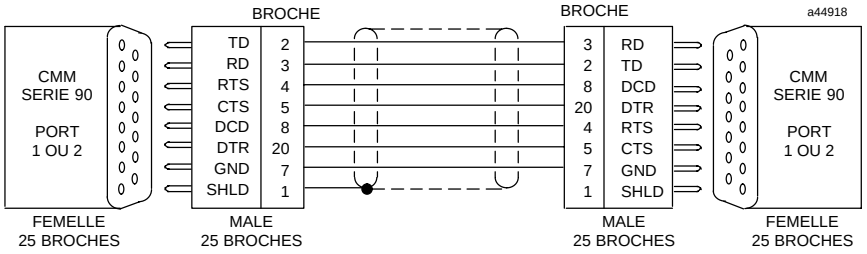


Figure 8-1. Module CMM à module CMM avec établissement d'une liaison (RS–232 seulement)

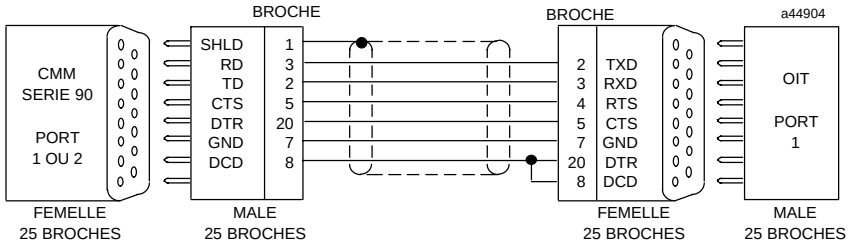


Figure 8-2. Module CMM à Terminal Opérateur d'Interface avec établissement d'une liaison (RS–232)

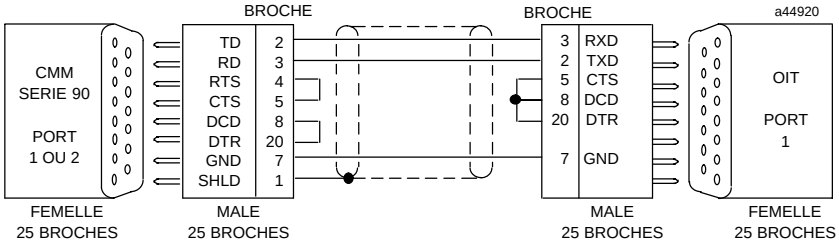


Figure 8-3. Module CMM à Terminal Opérateur d'Interface sans établissement d'une liaison (RS–232)

3. SCHÉMAS DE CÂBLAGE RS–422/RS–485

Vous trouverez ci–dessous des informations sur l'interface RS–485 et la fabrication des câbles RS–422/RS–485.

3.1. INTERFACE RS–422/RS–485 ET SCHÉMAS DE CÂBLAGE

3.1.1. L'interface RS–485

La gamme des API Série 90 satisfait les spécifications EIA RS–422 ou RS–485. Des circuits d'attaque et des récepteurs RS–485 sont utilisés pour faire communiquer plusieurs éléments du système via diverses combinaisons de gestionnaire/récepteur sur un seul câble à quatre paires torsadées. La longueur totale du câble ne doit pas excéder 1200 mètres.

Un système multipoint composé d'un émetteur et 8 récepteurs peut être configuré. La tension maximale en mode commun entre chaque point est la norme RS–485 de +12 Volts à –7 Volts. La sortie de l'émetteur doit fournir +/– 1,5 V minimum dans 60 ohms. L'impédance de sortie de l'émetteur doit être au moins de 120 Kohms à l'état Haut. La résistance du récepteur d'entrée est de 12 Kohms ou plus. La sensibilité du récepteur est de +/– 200 mV.

Attention

Toutes les précautions doivent être prises pour que les spécifications de tension en mode commun soient respectées. Les conditions en mode commun qui excèdent celles spécifiées génèreront des erreurs de transmission et/ou endommageront les éléments de la gamme des API Série 90.

3.1.1.1. Fabrication des câbles RS–422/RS–485

Si vous connectez le module CMM à un équipement autre qu'un API Série 90 à l'aide de l'interface RS–422/RS–485 standard, le module récepteur de l'équipement concerné doit être équipé d'une fonction "à sécurité intégrée". Cela signifie que dans un état Repos, ouvert, ou court–circuit, la sortie du module récepteur doit pouvoir adopter l'état "Mark" (signal à 1).

Avec l'interface RS–422/RS–485, les paires torsadées doivent être cablées de façon que les deux signaux d'émission forment une paire torsadée et les deux signaux de réception forment l'autre.

Le module CMM est fourni avec une résistance de terminaison de 120 ohms sur chaque circuit de réception RS–422. Si le module est à l'une ou l'autre extrémité d'une liaison multipoint ou point–à–point RS–422, la broche 25 (signal RD (B')) doit être directement reliée à la broche 24 (résistance de terminaison interne de 120 ohms). Si le module est un point intermédiaire dans une liaison multipoint, cette connexion n'a pas à être effectuée.

3.2. SCHÉMAS DE CÂBLAGE RS–422/RS–485

Vous trouverez dans ce paragraphe le schéma des connexions suivantes :

- Module CMM à module CMM
- Module CMM à Terminal Opérateur d'Interface
- Module CMM ou ordinateur–hôte à plusieurs modules CMM

Vous trouverez en outre les schémas correspondant aux connexions RS–422/RS–485 2 fils spéciales suivantes (disponibles sous protocole CCM uniquement) :

- Module CMM à module CMM (2 fils)
- Module CMM à plusieurs modules CMM (2 fils)

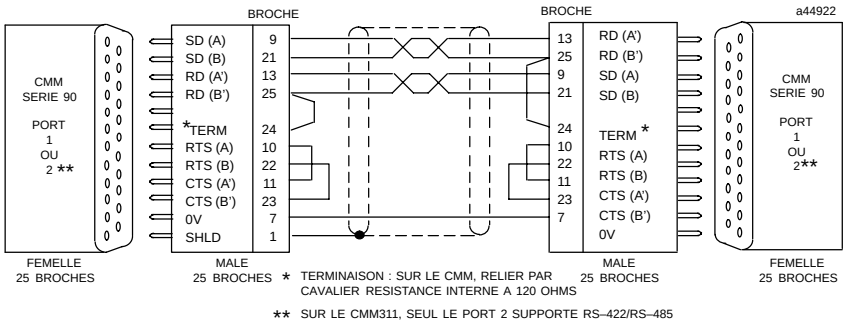


Figure 8-4. Module CMM à module CMM avec Contrl Flux = Aucun (RS–422/RS–485)

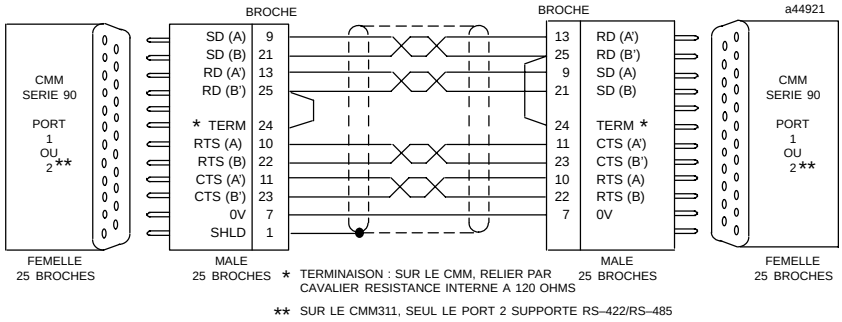


Figure 8-5. Module CMM à module CMM avec Contrl Flux = Matériel (RS–422/RS–485)

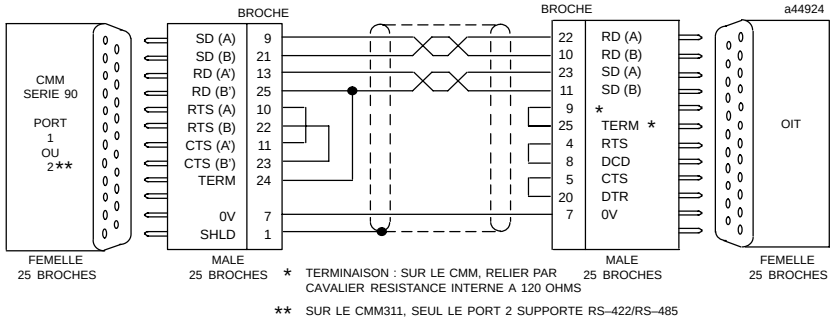


Figure 8-6. Module CMM à Terminal Opérateur d'Interface avec Contrl Flux = Aucun (RS–422/RS–485)

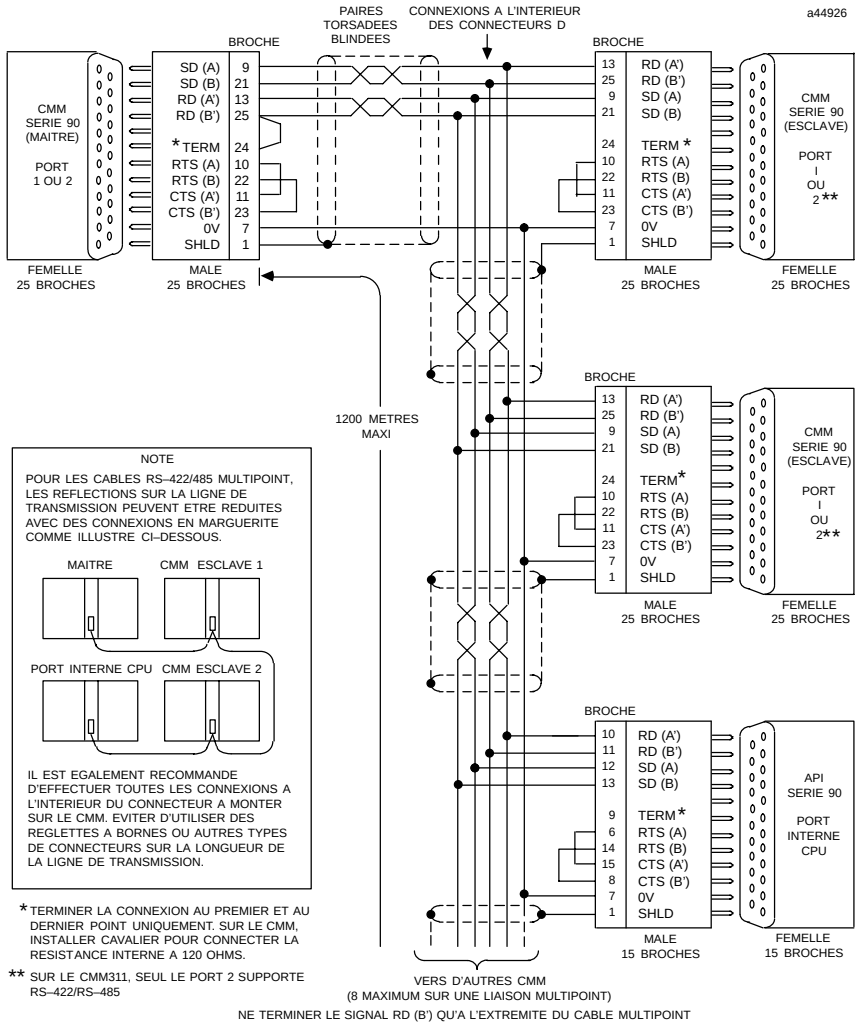

a44926
PAIRES TORSADEES BLINDEES
CONNEXIONS A L'INTERIEUR DES CONNECTEURS D
CMM SERIE 90 (MAITRE)
PORT 1 OU 2
FEMELLE 25 BROCHES
BROCHE
SD (A) 9
SD (B) 21
RD (A') 13
RD (B') 25
*TERM 24
RTS (A) 10
RTS (B) 22
CTS (A') 11
CTS (B') 23
0V 7
SHLD 1
MALE 25 BROCHES
1200 METRES MAXI
CMM SERIE 90 (ESCLAVE)
PORT I OU 2**
FEMELLE 25 BROCHES
RD (A') 13
RD (B') 25
SD (A) 9
SD (B) 21
TERM* 24
RTS (A) 10
RTS (B) 22
CTS (A') 11
CTS (B') 23
0V 7
SHLD 1
MALE 25 BROCHES
API SERIE 90
PORT INTERNE CPU
FEMELLE 15 BROCHES
RD (A') 10
RD (B') 11
SD (A) 12
SD (B) 13
TERM* 9
RTS (A) 6
RTS (B) 14
CTS (A') 15
CTS (B') 8
0V 7
SHLD 1
MALE 15 BROCHES
NOTE
POUR LES CABLES RS–422/485 MULTIPOINT, LES REFLECTIONS SUR LA LIGNE DE TRANSMISSION PEUVENT ETRE REDUITES AVEC DES CONNEXIONS EN MARGUERITE COMME ILLUSTRE CI–DESSOUS.
MAITRE
CMM ESCLAVE 1
PORT INTERNE CPU
CMM ESCLAVE 2
IL EST EGALEMENT RECOMMANDE D'EFFECTUER TOUTES LES CONNEXIONS A L'INTERIEUR DU CONNECTEUR A MONTER SUR LE CMM. EVITER D'UTILISER DES REGLETTES A BORNES OU AUTRES TYPES DE CONNECTEURS SUR LA LONGUEUR DE LA LIGNE DE TRANSMISSION.
*TERMINER LA CONNEXION AU PREMIER ET AU DERNIER POINT UNIQUEMENT. SUR LE CMM, INSTALLER CAVALIER POUR CONNECTER LA RESISTANCE INTERNE A 120 OHMS.
** SUR LE CMM311, SEUL LE PORT 2 SUPPORTE RS–422/RS–485
VERS D'AUTRES CMM (8 MAXIMUM SUR UNE LIAISON MULTIPOINT)
NE TERMINER LE SIGNAL RD (B') QU'A L'EXTREMITE DU CABLE MULTIPOINT

Figure 8-7. Module CMM ou ordinateur–hôte à plusieurs modules CMM (Multipoint 4 fils)

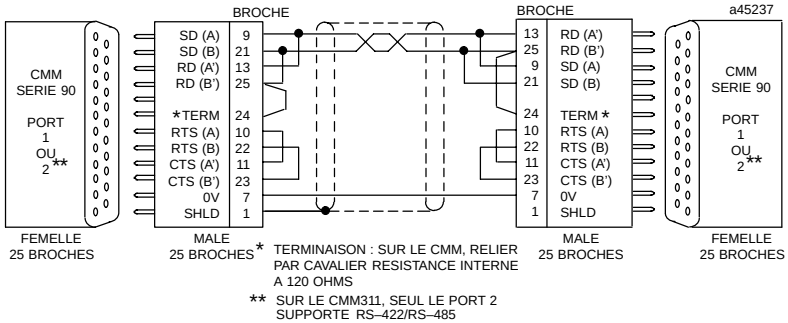


Figure 8-8. Module CMM à module CMM avec Contrl Flux = Aucun (RS–422/RS–485 2 fils)

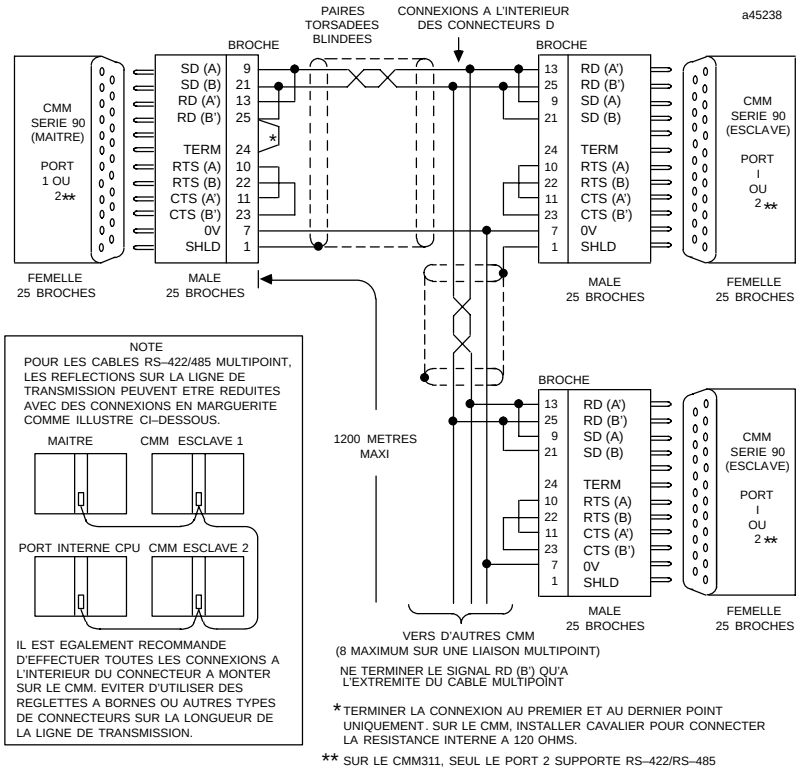

a45238
PAIRES TORSADEES BLINDEES
CONNEXIONS A L'INTERIEUR DES CONNECTEURS D
BROCHE
CMM SERIE 90 (MAITRE)
PORT 1 OU 2**
SD (A) 9
SD (B) 21
RD (A') 13
RD (B') 25
TERM 24
RTS (A) 10
RTS (B) 22
CTS (A') 11
CTS (B') 23
0V 7
SHLD 1
*
FEMELLE 25 BROCHES
MALE 25 BROCHES
BROCHE
RD (A') 13
RD (B') 25
SD (A) 9
SD (B) 21
TERM 24
RTS (A) 10
RTS (B) 22
CTS (A') 11
CTS (B') 23
0V 7
SHLD 1
MALE 25 BROCHES
CMM SERIE 90 (ESCLAVE) PORT I OU 2 **
FEMELLE 25 BROCHES
1200 METRES MAXI
BROCHE
RD (A') 13
RD (B') 25
SD (A) 9
SD (B) 21
TERM 24
RTS (A) 10
RTS (B) 22
CTS (A') 11
CTS (B') 23
0V 7
SHLD 1
MALE 25 BROCHES
CMM SERIE 90 (ESCLAVE) PORT I OU 2 **
FEMELLE 25 BROCHES
VERS D'AUTRES CMM
(8 MAXIMUM SUR UNE LIAISON MULTIPOINT)
NE TERMINER LE SIGNAL RD (B') QU'A L'EXTREMITE DU CABLE MULTIPOINT
NOTE
POUR LES CABLES RS–422/485 MULTIPOINT, LES REFLECTIONS SUR LA LIGNE DE TRANSMISSION PEUVENT ETRE REDUITES AVEC DES CONNEXIONS EN MARGUERITE COMME ILLUSTRE CI–DESSOUS.
MAITRE
CMM ESCLAVE 1
PORT INTERNE CPU
CMM ESCLAVE 2
IL EST EGALEMENT RECOMMANDE D'EFFECTUER TOUTES LES CONNEXIONS A L'INTERIEUR DU CONNECTEUR A MONTER SUR LE CMM. EVITER D'UTILISER DES REGLETTES A BORNES OU AUTRES TYPES DE CONNECTEURS SUR LA LONGUEUR DE LA LIGNE DE TRANSMISSION.
* TERMINER LA CONNEXION AU PREMIER ET AU DERNIER POINT UNIQUEMENT. SUR LE CMM, INSTALLER CAVALIER POUR CONNECTER LA RESISTANCE INTERNE A 120 OHMS.
** SUR LE CMM311, SEUL LE PORT 2 SUPPORTE RS–422/RS–485

Figure 8-9. Module CMM à plusieurs modules CMM (Multipoint 2 fils)

4. LE RÉPÉTEUR/CONVERTISSEUR ISOLÉ (IC655CCM590)

Vous trouverez ci–dessous des informations concernant l'utilisation du répéteur/convertisseur isolé dont :

- Description du répéteur/convertisseur isolé
- Configurations système
- Schémas de câblage

Cet élément est en vente chez GE Fanuc Automation. Pour plus d'informations, prenez contact avec votre agence commerciale ou votre contact GE Fanuc Automation.

4.1. DESCRIPTION DU RÉPÉTEUR/CONVERTISSEUR ISOLÉ

Le répéteur/convertisseur isolé (IC655CCM590) peut être utilisé dans les cas suivants :

- Pour fournir l'isolation de masse lorsqu'aucune masse commune ne peut être établie entre les composants.
- Pour amplifier les signaux RS–422 lorsque les distances sont plus grandes et qu'il y a plus de points de connexion.
- Pour convertir les signaux de RS–232 à RS–422 ou de RS–422 à RS–232.

Le schéma ci–dessous montre l'élément et la position des composants auxquels l'utilisateur peut accéder :

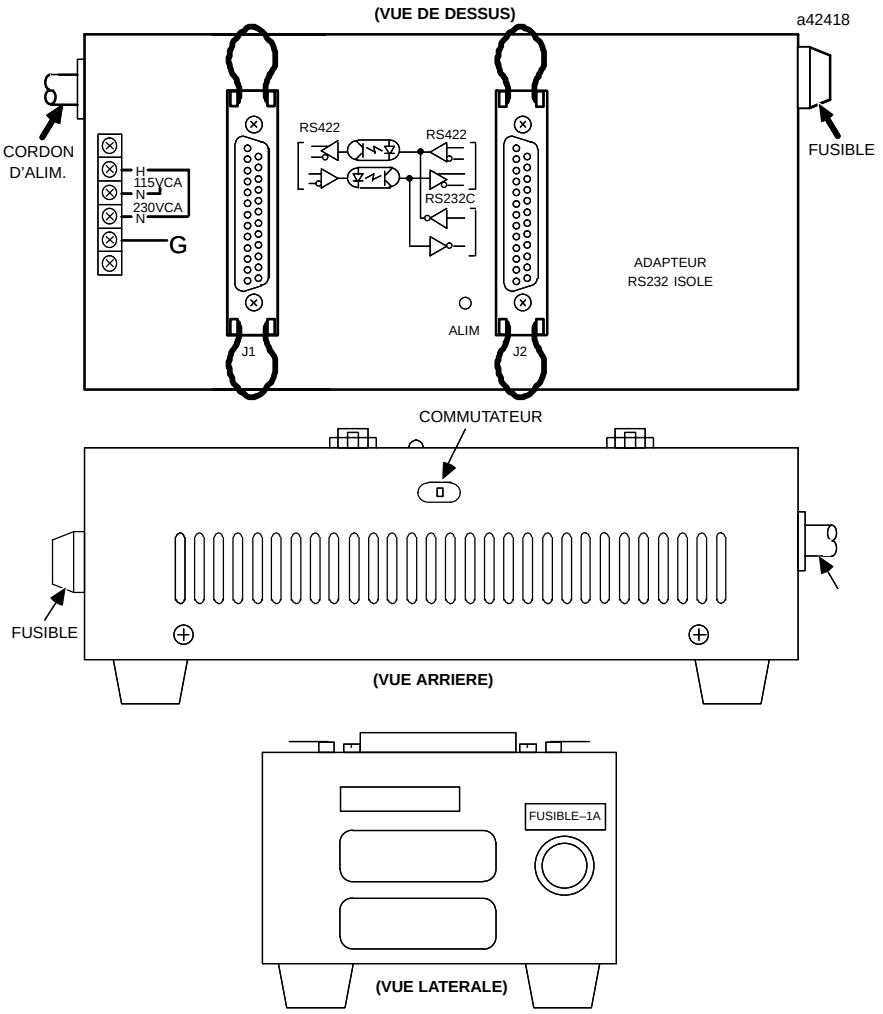


Figure 8-10. Répéteur/convertisseur isolé

Les composants du répéteur/convertisseur isolé auxquels l'utilisateur peut accéder sont décrits ci–dessous :

- Deux connecteurs femelles 25 broches Type D.

  (Deux connecteurs mâles 25 broches Type D (à souder) sont également fournis pour la fabrication des câbles.)

- Bornier 4 positions (interne) pour le raccordement d'une alimentation 115/230 Vca.
- Fusible de protection 1 A.
- Voyant de mise sous tension (vert).
- Le commutateur à levier 3 positions, placé sur la façade arrière, est à configurer conformément aux "Configurations système" décrites plus loin dans ce paragraphe.

4.1.1. Schéma de principe du répéteur/convertisseur isolé

Le schéma ci–dessous montre comment fonctionne l'élément. Vous remarquerez le commutateur à 3 positions qui commande les émetteurs du port J1. Ce commutateur est décrit dans "Configurations système", plus loin dans ce paragraphe.

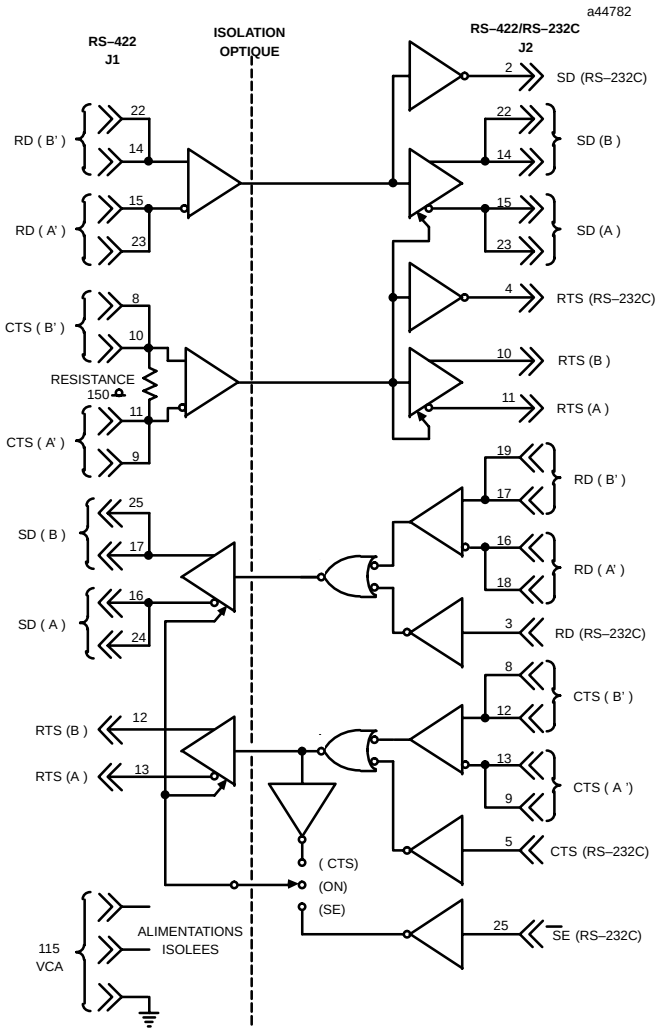


Figure 8-11. Schéma de principe du répéteur RS–422/convertisseur RS–232 isolé

Remarque

Toutes les entrées sur l'élément sont polarisées sur l'état Inactif. Les entrées non connectées génèreront un 1 binaire (état OFF) sur la sortie correspondante.

4.1.1.1. Affectation des broches du répéteur/convertisseur isolé

Port RS–422 J1 (connecteur femelle 25 broches)			Port RS–422/RS–232 J2 (connecteur femelle 25 broches)		
1		NC	1		NC
2		NC	2	SD	Emission de données (RS–232)
3		NC	3	RD	Réception de données (RS–232)
4		NC	4	RTS	Demande pour émettre (RS–232)
5		NC	5	CTS	Prêt à émettre (RS–232)
6		NC	6		NC
7	0V	Terre de signalisation	7	0V	Terre de signalisation
8	CTS(B')	Prêt à émettre (Terminaison optionnelle)	8	CTS(B')	Prêt à émettre (Terminaison optionnelle)
9	CTS(A')	Prêt à émettre (Terminaison optionnelle)	9	CTS(A')	Prêt à émettre (Terminaison optionnelle)
10	CTS(B')	Prêt à émettre	10	RTS(B)	Demande pour émettre
11	CTS(A')	Prêt à émettre	11	RTS(A)	Demande pour émettre
12	RTS(B)	Demande pour émettre	12	CTS(B')	Prêt à émettre
13	RTS(A)	Demande pour émettre	13	CTS(A')	Prêt à émettre
14	RD(B')	Réception de données	14	SD(B)	Emission de données
15	RD(A')	Réception de données	15	SD(A)	Emission de données
16	SD(A)	Emission de données	16	RD(A')	Réception de données
17	SD(B)	Emission de données	17	RD(B')	Réception de données
18		NC	18	RD(A')	Réception de données (Terminaison optionnelle)
19		NC	19	RD(B')	Réception de données (Terminaison optionnelle)
20		NC	20		NC
21		NC	21		NC
22	RD(B')	Réception de données (Terminaison optionnelle)	22	SD(B)	Emission de données
23	RD(A')	Réception de données (Terminaison optionnelle)	23	SD(A)	Emission de données
24	SD(A)	Emission de données	24		NC
25	SD(B)	Emission de données	25	SE	Valider (RS–232C) Non utilisée

NC = Non Connecté
(A) et (B) sont identiques pour – et +. A et B représentent les sorties, et A' et B' représentent les entrées.

Attention

Les connexions à la Terre de signalisation (broche 7 de chaque connecteur) doivent être effectuées entre le répéteur/convertisseur isolé et l'API pour J1, et entre le répéteur/convertisseur isolé et l'ordinateur–hôte pour J2.

La broche 7 du port J1 est reliée sur l'enveloppe métallique du connecteur J1 et la broche 7 du port J2 à l'enveloppe métallique du connecteur J2. Ces deux connexions à la terre de signalisation sont isolées l'une de l'autre et sont également isolées de la masse du système d'alimentation (fil vert sur le bornier). Pour que l'isolation soit préservée, ces terres de signalisation ne doivent pas être reliées ensemble.

4.2. CONFIGURATIONS SYSTÈME

Les schémas ci–après montrent les différentes méthodes que vous pouvez utiliser pour connecter le répéteur/convertisseur isolé pour convertir des signaux, augmenter le nombre de points ou la distance. Toute configuration système peut être réduite à un nombre minimum de câbles, chacun couvrant une partie de la configuration système globale. Dans les configurations système indiquées ci–après à titre d'exemple, ces câbles sont notés Câbles A–D et sont illustrés dans les "Schémas de câblage" qui suivent.

Conflit d'accès descendant et ascendant. Dans ce paragraphe, on appelera configurations multipoint simples les configurations qui n'utilisent qu'un seul répéteur/convertisseur isolé. Les configurations multipoint complexes comptent une ou plusieurs sections multipoint dans lesquelles un répéteur/convertisseur isolé fait office de point. Dans toutes les configurations (simples ou complexes), les émetteurs vers l'aval depuis l'équipement maître peuvent toujours être actifs. Il n'y aura pas de conflit d'accès à la ligne de communication dans la mesure où un seul équipement (l'équipement maître) peut transmettre dans le sens descendant.

Dans les configurations multipoint simples, il n'y aura pas de conflit d'accès dans le sens ascendant tant que les équipements laissent leurs circuits d'attaque dans l'état à haute impédance lorsqu'ils sont inactifs et ne les activent que lorsqu'ils ont un message à transmettre. C'est le cas des modules CMM Série 90–70 et Série 90–30.

Dans les configurations multipoint complexes, toutefois, des dispositions spéciales doivent être prises pour commuter les émetteurs dans le sens ascendant du répéteur/convertisseur isolé.

Commutation des émetteurs vers l'amont. Pour que les circuits d'attaque RS–422 soit actifs sur le port J2 du répéteur/convertisseur isolé, l'entrée RTS sur J1 doit être à vrai. L'état des circuits d'attaque RS–422 sur le port J1 dépend de la position du commutateur sur l'unité. Lorsque le commutateur est positionné au centre, les émetteurs J1 seront toujours actifs. Lorsque le commutateur est en position CTS, (vers le cordon d'alimentation), le signal CTS RS–232 ou RS–422 CTS doit être à VRAI pour activer les circuits d'attaque J1.

Remarque

Notez la position du commutateur du répéteur/convertisseur isolé dans les configurations système ci–après.

4.2.1. Configuration multipoint simple

Cette configuration montre comment connecter un répéteur/convertisseur isolé pour convertir des signaux ou augmenter la distance.

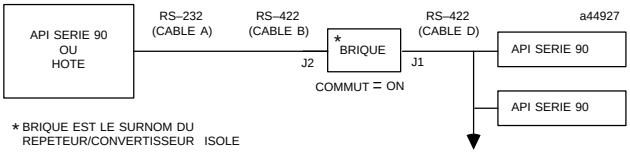


Figure 8-12. Configuration système simple avec un seul répéteur/convertisseur isolé

4.2.1.1. Configuration multipoint complexe

Cette configuration montre comment connecter plusieurs répéteurs/convertisseurs isolés pour convertir des signaux, augmenter la distance et le nombre de points.

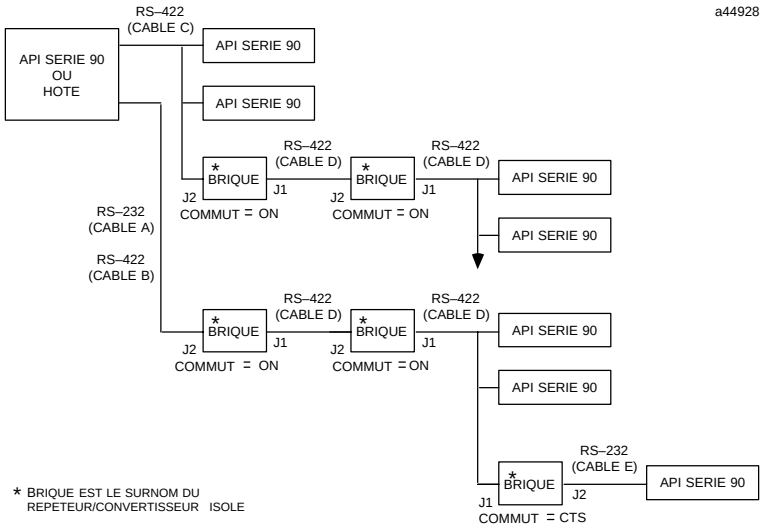


Figure 8-13. Configuration système complexe avec plusieurs répéteurs/convertisseurs isolés

4.2.1.2. Règles d'utilisation des répéteurs/convertisseurs dans des réseaux complexes

Dans les configurations de réseau multipoint complexe comportant plusieurs API et répéteurs/convertisseurs RS–422 (désignés briques par la suite), les règles suivantes s'appliquent :

Règle 1 : Lorsqu'une brique est utilisée comme répéteur, le port J2 doit toujours être acheminé vers l'équipement–hôte, et le port J1 dans la direction inverse. Le commutateur situé sur la façade latérale de la brique doit être positionné au centre (ON). Le seul cas dans lequel le port 1 est acheminé vers l'hôte est lorsque la brique est utilisée comme convertisseur (RS–232) sur l'équipement esclave. Le commutateur doit alors être positionné à droite (CTS).

Règle 2 : Si l'équipement esclave d'un module CMM Série 90 se trouve en aval d'une brique, configurez le port série du module CMM sur AUCUN contrôle de flux avec un délai de retournement (Délai TurnA) de 10 ms. (Ne s'applique qu'aux protocoles CCM, SNP, et SNP–X).

Règle 3 : N'insérez pas plus de 3 briques sur une voie de transmission de données entre l'ordinateur–hôte et les équipements esclaves.

4.3. SCHÉMAS DE CÂBLAGE

Les schémas de câblage ci–dessous font référence aux câbles A–D identifiés dans les configurations système de la page précédente. Ces schémas montrent le principe de fabrication des câbles et peuvent être modifiés en fonction des besoins de votre application spécifique.

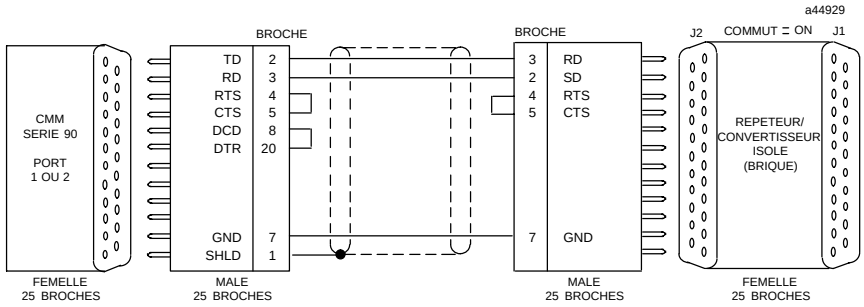


Figure 8-14. Câble A ; CMM RS–232 à convertisseur

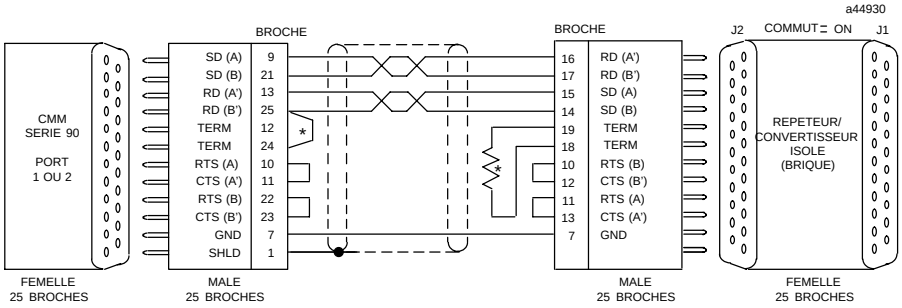
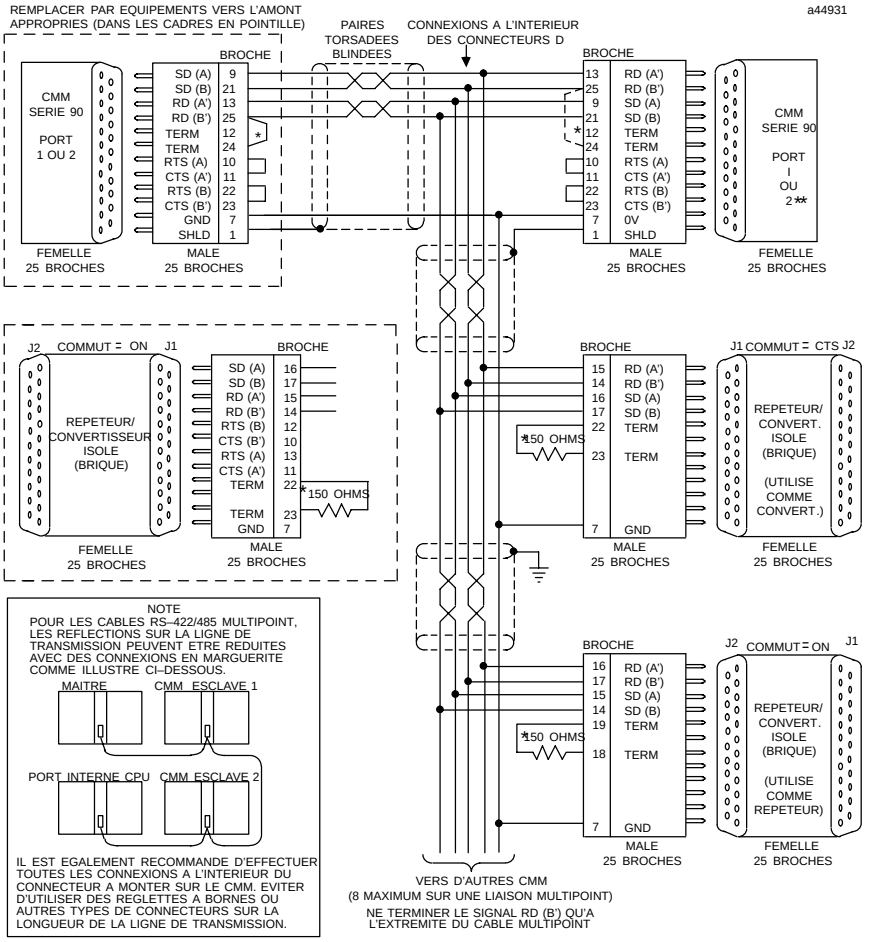


Figure 8-15. Câble B ; CMM RS–422 à convertisseur



* TERMINER LA CONNEXION AU PREMIER ET AU DERNIER POINT UNIQUEMENT. SUR LE CMM, INSTALLER UN CAVALIER POUR CONNECTER LA RESISTANCE INTERNE A 120 OHMS. SUR LE REPETEUR/CONVERTISSEUR ISOLE, INSTALLER LA RESISTANCE A 150 OHMS (FOURNIE).

** SUR LE CMM311, SEUL LE PORT 2 SUPPORTE RS–422/RS–485

Figure 8-16. Câble C ; Paire torsadée RS422

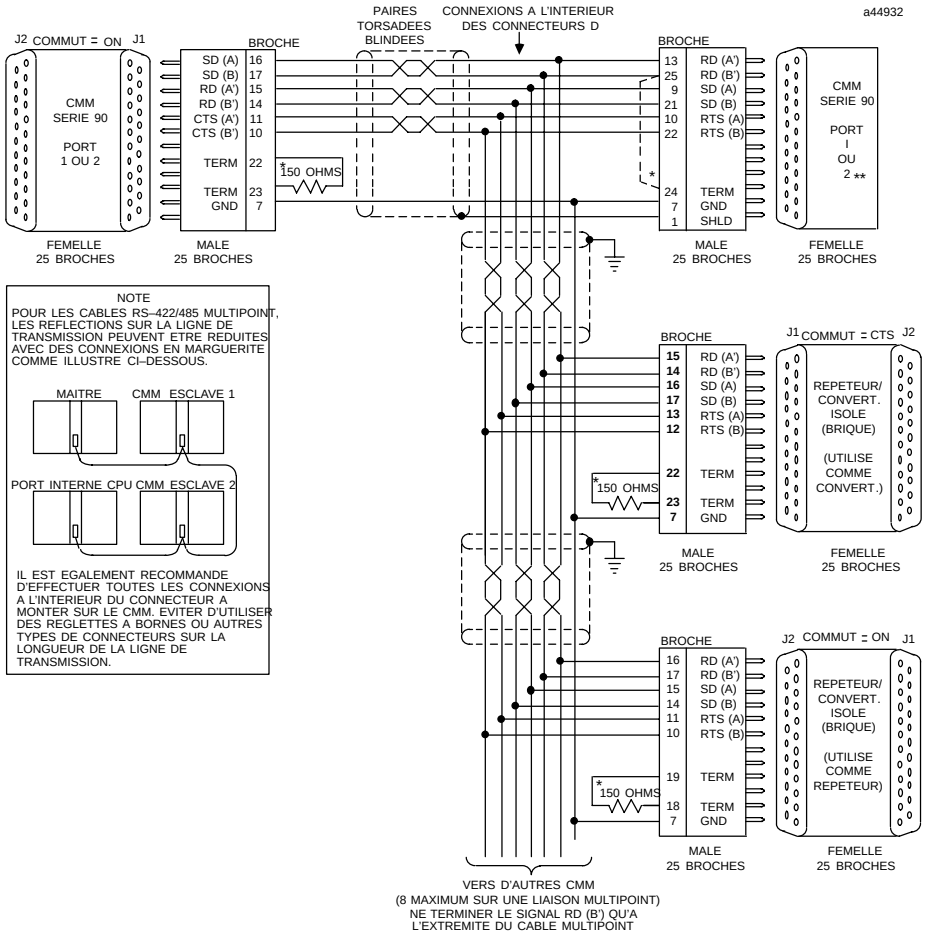

a44932
PAIRES TORSADEES BLINDEES
CONNEXIONS A L'INTERIEUR DES CONNECTEURS D
J2 COMMUT = ON J1
CMM SERIE 90
PORT 1 OU 2
BROCHE
SD (A) 16
SD (B) 17
RD (A') 15
RD (B') 14
CTS (A') 11
CTS (B') 10
TERM 22
TERM 23
GND 7
150 OHMS
FEMELLE 25 BROCHES
MALE 25 BROCHES
BROCHE
13 RD (A')
25 RD (B')
9 SD (A)
21 SD (B)
10 RTS (A)
22 RTS (B)
24 TERM
7 GND
1 SHLD
MALE 25 BROCHES
CMM SERIE 90
PORT I OU 2 **
FEMELLE 25 BROCHES
NOTE
POUR LES CABLES RS–422/485 MULTIPOINT, LES REFLECTIONS SUR LA LIGNE DE TRANSMISSION PEUVENT ETRE REDUITES AVEC DES CONNEXIONS EN MARGUERITE COMME ILLUSTRE CI–DESSOUS.
MAITRE
CMM ESCLAVE 1
PORT INTERNE CPU CMM ESCLAVE 2
IL EST EGALEMENT RECOMMANDE D'EFFECTUER TOUTES LES CONNEXIONS A L'INTERIEUR DU CONNECTEUR A MONTER SUR LE CMM. EVITER D'UTILISER DES REGLETTES A BORNES OU AUTRES TYPES DE CONNECTEURS SUR LA LONGUEUR DE LA LIGNE DE TRANSMISSION.
BROCHE
15 RD (A')
14 RD (B')
16 SD (A)
17 SD (B)
13 RTS (A)
12 RTS (B)
22 TERM
23 TERM
7 GND
150 OHMS
MALE 25 BROCHES
J1 COMMUT = CTS J2
REPETEUR/ CONVERT. ISOLE (BRIQUE)
(UTILISE COMME CONVERT.)
FEMELLE 25 BROCHES
BROCHE
16 RD (A')
17 RD (B')
15 SD (A)
14 SD (B)
11 RTS (A)
10 RTS (B)
19 TERM
18 TERM
7 GND
150 OHMS
MALE 25 BROCHES
J2 COMMUT = ON J1
REPETEUR/ CONVERT. ISOLE (BRIQUE)
(UTILISE COMME REPETEUR)
FEMELLE 25 BROCHES
VERS D'AUTRES CMM
(8 MAXIMUM SUR UNE LIAISON MULTIPOINT)
NE TERMINER LE SIGNAL RD (B') QU'A L'EXTREMITE DU CABLE MULTIPOINT
* TERMINER LA CONNEXION AU PREMIER ET AU DERNIER POINT UNIQUEMENT. SUR LE CMM, INSTALLER UN CAVALIER POUR CONNECTER LA RESISTANCE INTERNE A 120 OHMS. SUR LE REPETEUR/CONVERTISSEUR ISOLE, INSTALLER LA RESISTANCE A 150 OHMS (FOURNIE).
** SUR LE CMM311, SEUL LE PORT 2 SUPPORTE RS–422/RS–485

Figure 8-17. Câble D ; Paire torsadée RS422

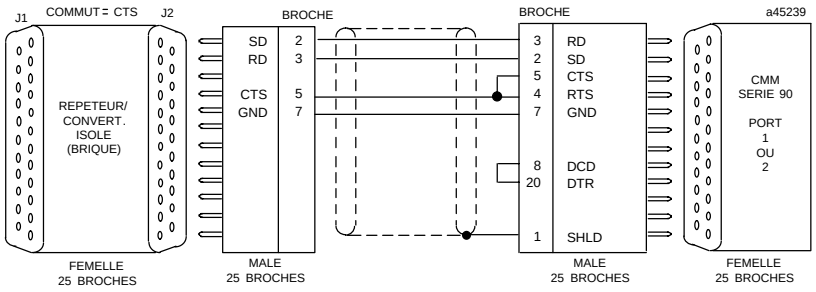


Figure 8-18. Câble E ; Convertisseur RS–232 à CMM

Annexe A Glossaire

Le vocabulaire relatif à la communication série des API de la gamme Série 90 est très particulier et de nombreux termes sont désignés par des acronymes. Par exemple, un Automate Programmable Industriel (API), un ordinateur ou tout autre équipement connecté à un réseau est désigné sous le terme générique *station*.

Cette annexe contient une liste alphabétique concise des termes couramment employés dans le langage des communications et, s'ils existent, des acronymes correspondant. La plupart de ces termes sont utilisés dans ce manuel.

1. ACRONYMES ET ABRÉVIATIONS COURANTES

API	Automate Programmable Industriel.
ASCII	(American national Standard Code for Information Interchange) Code de transmission de données américain.
ACK	Caractère d'accusé de réception.
BCC	Code de contrôle de bloc.
BCD	Décimal Codé Binaire.
BEM	(Bus Expansion Module) Module d'extension de bus (BTM ou BRM).
BPS	Bits Par Seconde.
BTM	(Bus Transmitter Module) Module émetteur sur bus.
CCM	Protocole utilisé par le module coprocesseur de communication des API de la gamme Série 90.
CMM	Abréviation de la référence produit du module coprocesseur de communication. Utilisé pour désigner le module.
COMREQ	COMmunication REQuest (demande de communication).
CRC	Contrôle de redondance cyclique
CTS	Signal de commande Prêt à émettre
DCD	Signal de commande Détection de transmission de données.
DMA	Accès direct à la mémoire
DOS	Système d'exploitation de disque.
DTR	Signal de commande Terminal de données prêt.
EEPROM	(Electrically Erasable Programmable Read Only Memory) Mémoire programmable effaçableélectriquement.
EIA	Association des Industries d'Electronique (USA).
ENQ	Caractère de commande Demande.
EOT	Caractère de commande Fin de transmission.
EPROM	(Erasable Programmable Read Only Memory) Mémoire morte programmable effaçable.
E/S	Entrées/Sorties.
ETB	Caractère de commande Fin de bloc
ETCD	Equipement terminal de circuit de données.
ETTD	Equipement terminal de transmission de données
ETX	Caractère de commande Fin de texte.
FCC	(Federal Communications Commission) Commisssion fédérale des télécommunications (USA).
H	Hexadécimal.
IEEE	Institute of Electrical and Electronics Engineers
ISO	Organisation internationale de normalisation.
K	1024.
Ko	Kilo–octet (1024 octets).
LRC	Contrôle de redondance longitudinale.
MAP	(Manufacturing Automation Protocol) Protocole d'automatisation de fabrication.
Mo	Mega–octet (1 048 576 octets).
MODEM	MOdulateur/DEModulateur.
NAK	Caractère de commande Accusé de réception négatif.
OIT	(Operator Interface Terminal) Terminal opérateur
PC	Ordinateur personnel.
PCM	Module Coprocesseur Programmable
PROM	(Programmable Read Only Memory) Mémoire morte programmable.
RAM	(Random Access Memory) Mémoire vive.
RD, RXD	Signal de réception de données.
RTS	Signal de commande Prêt à émettre.
RTU	Protocole Remote Terminal Unit
SD, TXD	Signal de transmission de données.
SNP	(Series Ninety Protocol) Protocole SNP
SNP–X	Protocole SNP étendu.
SOH	Caractère de commande Début d'en–tête.
STX	Caractère de commande Début de texte.
UC	Unité centrale
VME	Versa Module European

2. GLOSSAIRE

Actionneur

Equipements physiques tels que démarreur de moteur, charge inductive, etc., actionnés par l'API.

Adresse

Série de nombres décimaux affectés à des emplacements mémoires spécifiques du programme et utilisés pour accéder à ces emplacements.

Analogique

Expression numérique de variables physiques telles que des rotations, des distances et des tensions afin de représenter une quantité.

API (Automate Programmable Industriel)

Dispositif de contrôle industriel électronique. L'API reçoit des signaux d'équipements de commande fournis par l'utilisateur (interrupteurs, capteurs), les traite en suivant un modèle précis déterminé par des programmes à diagramme en échelle contenus dans la mémoire utilisateur, et fournit des sorties pour le contrôle de procédés ou d'équipements fournis par l'utilisateur tels que les relais ou les démarreurs de moteurs.

ASCII

Acronyme pour American Standard Code for Information Interchange. Code sur 8 bits (7 bits plus 1 bit de parité) utilisé pour représenter les données.

Asynchrone

Transmission de données autorisant des intervalles de temps de longueurs différentes entre les caractères transmis. La transmission asynchrone est contrôlée par des bits de départ et des bits de stop, au début et à la fin de chaque caractère.

Baud

Unité de transmission de données. Le débit (exprimé en bauds) correspond au nombre d'états discrets (de bits dans certaines conditions) transmis par seconde.

Binaire

Système de numérotation utilisant uniquement les chiffres 0 et 1. Egalement appelé base 2.

Bit

Plus petite unité de mémoire. Elle permet le stockage d'un élément d'information à deux valeurs possibles (un/zéro; on/off, bon/mauvais, oui/non). Les données représentant plus de deux valeurs (par exemple les valeurs numériques comprises entre 000 et 999) sont représentées par un ensemble de plusieurs bits.

Bit de parité

Bit ajouté à un mot mémoire pour que la somme de ses bits soit paire (parité paire) ou impaire (parité impaire).

Bit significatif

Bit contribuant à la précision d'un nombre. Le nombre de bits significatifs est compté à partir du bit contribuant à la valeur la plus grande, appelé bit de poids fort (BPF), jusqu'au bit contribuant à la valeur la moins grande, appelé bit de poids faible (BPf).

Break

Signal de ”réveil” transmis sur la ligne série et précédant chaque nouvelle session de communication SNP. (Voir chapitre 7, § 2, Protocole SNP, pour plus de détails.)

Brique

Mnémonique pour le répéteur/convertisseur isolé RS–232/RS–485 de GE Fanuc.

Bruit

Perturbations électriques gênant les signaux normaux et généralement de haute fréquence.

Bus

Chemin électrique permettant le transfert et la réception de données.

Câble en ”Y”

Câble fourni avec chaque module CMM Série 90–30 qui sépare les ports série 1 et 2 du port série combiné du module CMM.

Communication série

Méthode de transfert de données qui consiste à envoyer les bits en séquence, et non simultanément comme dans la transmission parallèle.

Connexion

Message SNP établissant la communication entre l’équipement SNP maître et un équipement SNP esclave spécifique. (Voir chapitre 7, § 3, Protocole SNP, pour plus de détails.)

Connexion longue

Message SNP, suivi d’un message Sélectionner les paramètres SNP. Utilisé pour établir une communication SNP et pour négocier les paramètres de fonctionnement normal entre l’équipement SNP maître et un équipement SNP esclave spécifique. (Voir chapitre 7, § 3, Protocole SNP, pour plus de détails.)

Console de programmation

Equipement permettant l’entrée, l’examen et la modification de la mémoire de l’API, y compris les zones de programme et de stockage.

Contrôle de parité

Contrôle qui détermine si le nombre total de ”1” dans un mot est pair ou impair.

Coupleur de scrutation des E/S

Méthode utilisée par l’UC pour surveiller toutes les entrées et pour contrôler toutes les sorties pendant un temps donné.

Diagramme en échelle

Représentation d’un système de contrôle commande. Le programme utilisateur est exprimé avec des symboles équivalents à des relais.

Ecrire

Opération de transfert, d’enregistrement ou de copie de données d’une mémoire à une autre.

Entrée

Signal, généralement ”0” ou ”1”, qui fournit des informations à l’API. Les entrées sont habituellement générées par des équipements tels que des interrupteurs de fin de course ou des boutons–poussoirs.

Erreur de parité

Erreur qui se produit lorsqu’un contrôle de parité (checksum) n’est pas conforme au bit de parité.

Egal–à–égal

Communication entre stations où toutes les stations peuvent émettre des demandes et y répondre.

Equipement périphérique

Equipements périphériques qui peuvent communiquer avec un API (par exemple consoles de programmation et imprimantes).

État complémentaire

Six octets d’informations générales relatives à l’API et renvoyés dans chaque message de réponse SNP. Ces données peuvent être transmises à l’API en cas de besoin.

E/S (Entrées/Sorties)

Partie de l’API à laquelle se raccordent les équipements sur site.

Ethernet

Technologie de réseau utilisant la méthode d’accès CSMA/CD (Carrier Source Multiple Access with Collision Detection).

Fenêtres de communication

Communication se déroulant au cours de la scutation de l’API entre le programme à diagramme en échelle et le module d’interface local.

Fond de bac

Groupe de connecteurs montés physiquement sur une carte située au dos d’un bac et pouvant recevoir des modules.

Hexadécimal

Système de numérotation en base 16, utilisant les chiffres 0 à 9 et les lettres A à F.

Interface

Permet de connecter un Automate Programmable Industriel à ses équipements d’application, canaux de communication et autres équipements à travers différents câbles et modules.

Isolation

Méthode de séparation du câblage sur site et des circuits logiques, généralement mise en oeuvre avec des périphériques d’isolation optique à semi–conducteurs.

Isolation optique

Utilisation d’un dispositif à semi–conducteurs pour isoler les capteurs et les actionneurs de l’utilisateur des circuits internes des modules d’E/S et de l’UC.

K (Kilo)

Abréviation pour kilo, qui équivaut exactement à 1024 dans le langage informatique. Généralement utilisée pour désigner 1024 mots mémoire.

Liaison de données

Equipement permettant la transmission des informations, y compris les modules d'interface et les câbles.

Lire

Extraire les données stockées dans une mémoire.

Maître–esclave

Communication entre stations où une seule station émet les demandes et où les autres stations ne font que répondre.

Matériel

Tous les équipements mécaniques, électriques et électroniques constituant un Automate Programmable Industriel et ses applications.

Mémoire

Groupement d'éléments de circuits physiques possédant des capacités d'entrée, de stockage et de récupération des données.

Mémoire non–volatile

Mémoire qui conserve les informations qu'elle contient en cas de coupure d'alimentation.

Mémoire utilisateur

Terme communément employée pour désigner les circuits de mémoire de l'API utilisés pour le stockage des programme en langage relais de l'utilisateur.

Mémoire volatile

Mémoire qui, sans alimentation, ne conserve pas les informations qu'elle contient.

Message Maintien de la liaison

Message généré par le maître ou l'esclave SNP ayant pour seul but d'éviter un dépassement de temporisation au niveau de l'équipement connecté (autrement dit, pour maintenir la communication SNP).

Microprocesseur

Processeur d'ordinateur électronique constitué de circuits intégrés contenant des fonctions arithmétiques et logiques ainsi que des fonctions de commande et de mémoire.

Microprogramme

Série d'instructions contenues en ROM (Read Only Memory) utilisées uniquement par les fonctions de traitement interne. Ces instructions sont transparentes pour l'utilisateur.

Microseconde (µs)

Un millionième de seconde. 1×10^{-6} ou 0,000 001 seconde.

Milliseconde (ms)

Un millième de seconde. 1 x 10^{-3} ou 0,001 seconde.

Mnémonique

Abréviation du nom d'une instruction. Les mnémoniques sont généralement des acronymes ou des sigles.

Mode autonome

Les équipements fonctionnant en mode autonome ne sont pas connectés à une ligne de communication. Par exemple, un ordinateur compatible PC en mode autonome travaille indépendamment de l'UC de l'API Série 90.

Mode connecté

Les équipements fonctionnant en mode connecté sont connectés à une ligne de communication.

Modèle de référence ISO pour l'interconnexion des systèmes ouverts

Standard international pour les architectures de réseau qui définit un modèle à sept couches. L'objectif est de fournir un guide de conception de réseau permettant de faire communiquer des équipements provenant de différents constructeurs.

Module

Sous–ensemble électronique remplaçable. Habituellement enfiché et fixé, il est très aisément remplaçable en cas de panne ou de modification du système. Pour les API de la gamme Série 90, c'est un ensemble constitué d'une carte à circuit imprimé et d'un plastron.

Module coprocesseur de communication (CMM)

Le module coprocesseur de communication fournit une interface série entre l'API Série 90–70 ou 90–30 et d'autres abonnés sur le réseau. Les modules CMM supportent les protocoles CCM, RTU et SNP.

Module d'entrée

Module d'E/S qui convertit les signaux transmis par les équipements utilisateur en niveaux logiques utilisables par l'UC.

Module de sortie

Module d'E/S qui convertit les signaux logiques de l'UC en signaux de sortie utilisables pour contrôler une machine ou un procédé.

Mot

Mesure de longueur de mémoire, habituellement 16 bits (pour les API de la gamme Série 90).

Mot d'état

Indique l'état global des protocoles CCM et SNP et du réseau de communication.

Mots d'état de diagnostic

Groupe de 20 mots fournissant des informations détaillées sur le fonctionnement et la configuration du module CMM et utilisés pour la surveillance et le diagnostic des erreurs de transmission. Les mots d'état sont suivis et mis à jour dans le module CMM. Les protocoles CCM et SNP utilisent des ensembles de mots d'état de diagnostic différents.

Multipoint

Configuration de câblage série reliant plus de deux équipements. En configuration multipoint, tous les équipements doivent posséder une adresse unique.

Nanoseconde (ns)

Un milliardième de seconde. 1 x 10^{-9} ou 0,000 000 001 seconde.

Normes ISO

L'organisation internationale de normalisation (ISO) pour l'interconnexion des systèmes ouverts (OSI).

Octet

Groupe de bits utilisé en tant qu'unité entière. Dans la gamme des API Série 90, un octet est composé de 8 bits.

Parité

L'état attendu, pair ou impair, d'un ensemble de chiffres binaires.

Point–à–point

Configuration de câblage série reliant uniquement deux équipements.

Programme

Séquence de fonctions entrée dans un Automate Programmable Industriel et exécutée par le processeur afin de contrôler une machine ou un procédé.

Programme d'application

Programme en diagramme en échelle exécuté dans un API, ou programme utilisateur exécuté dans un ordinateur.

Programme logique

Ensemble fixe de réponses (sorties) à différentes conditions externes (entrées). L'utilisateur doit spécifier toutes les situations possibles pour les activités, synchrones ou asynchrones. Egalement appelé programme.

PROM

Acronyme pour "Programmable Read Only Memory". La PROM est une mémoire morte programmée en usine que l'utilisateur ne peut pas modifier aisément.

Protocole

Ensemble de règles pour l'échange de messages entre deux processus communiquant.

Protocole MAP

Le protocole de communication MAP, spécifié par le protocole d'automatisation de fabrication, est un protocole "orienté connexion" ; autrement dit, les stations présentes sur un réseau MAP ne peuvent échanger des informations qu'après avoir établi une connexion logique, comme deux personnes communiquant par un système téléphonique.

Protocole SNP

Le protocole SNP est le protocole de communication natif des API de la gamme Série 90. SNP est un protocole propriétaire maître–esclave.

Protocole SNP étendu (SNP–X)

Extension optimisée du protocole SNP de base, utilisée uniquement pour augmenter la vitesse de transmission des données.

RAM

Acronyme pour ”Random Access Memory”. La RAM est une mémoire permettant de stocker et de récupérer directement des bits individuels. Ce type de mémoire est volatile ; autrement dit, les données en RAM sont perdues lorsque l’alimentation est interrompue. Il est nécessaire d’utiliser une pile de protection pour sauvegarder son contenu. Les API Série 90 utilisent une pile au lithium et au dioxyde de manganèse ou une pile de sauvegarde externe (optionnelle).

Référence

Nombre utilisé dans un programme pour indiquer à l’UC l’origine et la destination des données.

Réseau bande porteuse

Réseau conçu pour gérer des applications de taille petite ou moyenne avec 6–20 stations connectées.

Réseau large bande

Réseau conçu pour gérer des applications de taille moyenne ou grande avec plusieurs centaines de stations connectées. La technologie large bande est utilisée dans les systèmes de réseau importants et nécessite un remodulateur frontal.

RS–232

Standard spécifié par l’association des industries d’électronique (EIA) et relatif aux caractéristiques mécaniques et électriques de l’interface de connexion entre les équipements terminaux de transmission de données (ETTD) et les équipements terminaux de circuit de données (ETCD).

RS–422

Standard recommandé définissant les caractéristiques de l’interface électrique utilisée pour connecter un équipement terminal de transmission de données (ETTD) ou un équipement terminal de circuit de données (ETCD). Le standard RS–422 permet des distances et des vitesses de transmission supérieures à celles du stantard RS–232D.

RS–485

Identique à RS–422. Les circuits de réception bénéficient cependant de protections supplémentaires et d’une sensibilité améliorée permettant d’utiliser des distances et un nombre de branchements supérieurs.

RTU (Remote Terminal Unit)

Le protocole RTU, qui est du type demande–réponse, est utilisé pour la communication entre l’équipement RTU et l’ordinateur hôte. L’ordinateur hôte transmet la demande à l’esclave RTU qui peut uniquement répondre au maître.

Scrutation

Technique d’examen ou d’exécution en ordre séquentiel (de la première étape à la dernière) de toutes les étapes logiques spécifiées par le programme.

Segment

Dans un API, séquence ou groupement de fonctions qui contrôlent une bobine.

Séquence Q

Le format de protocole de séquence Q est utilisé pour demander et transférer 4 octets de données entre un esclave et un maître sans utiliser l'en–tête de 17 octets.

Sortie

Données sortant de l'UC et destinées à contrôler un équipement ou un procédé. Les données passent d'abord par un module de conversion de niveau.

Sorties

Signal, généralement "1" ou "0", transmis par l'API et dont l'alimentation est fournie par l'utilisateur, qui contrôle des équipements externes en fonction des commandes émises par l'UC.

Station initiatrice

Station à l'origine de la communication.

Stockage

Synonyme de mémoire.

Synchrone

Transmission dans laquelle les bits de données sont transmis à une vitesse constante, l'émetteur et le récepteur étant synchronisés par une horloge. La transmission synchrone évite l'utilisation des bits de départ et des bits de stop.

Table de références

L'une des zones mémoire adressables de l'API dans un programme en diagramme en échelle ou par le protocole SNP. La mémoire de l'API est adressée par type et par adresse.

Terminaison

Equipement (ou charge) connecté(e) à la sortie d'une ligne de transmission pour assurer la terminaison ou la fin des signaux de cette ligne.

UC (Unité Centrale)

Equipement central d'un API qui interprète les instructions utilisateur, prend les décisions et exécute les fonctions conformément à un programme d'application enregistré. Ce programme spécifie les actions à entreprendre pour chacune des entrées possibles.

Annexe B Liste des codes ASCII

Car.	Déc.	Hex.	Car.	Déc.	Hex.	Car.	Déc.	Hex.
NUL	0	00	0	48	30	[	91	5B
SOH	1	01	1	49	31	\	92	5C
STX	2	02	2	50	32	]	93	5D
ETX	3	03	3	51	33	^	94	5E
EOT	4	04	4	52	34	_	95	5F
ENQ	5	05	5	53	35	’	96	60
ACK	6	06	6	54	36	a	97	61
BEL	7	07	7	55	37	b	98	62
BS	8	08	8	56	38	c	99	63
HT	9	09	9	57	39	d	100	64
LF	10	0A	:	58	3A	e	101	65
VT	11	0B	;	59	3B	f	102	66
FF	12	0C	<	60	3C	g	103	67
CR	13	0D	=	61	3D	h	104	68
SO	14	0E	>	62	3E	i	105	69
SI	15	0F	?	63	3F	j	106	6A
DLE	16	10	@	64	40	k	107	6B
DC1	17	11	A	65	41	l	108	6C
DC2	18	12	B	66	42	m	109	6D
DC3	19	13	C	67	43	n	110	6E
DC4	20	14	D	68	44	o	111	6F
NAK	21	15	E	69	45	p	112	70
SYN	22	16	F	70	46	q	113	71
ETB	23	17	G	71	47	r	114	72
CAN	24	18	H	72	48	s	115	73
EM	25	19	I	73	49	t	116	74
SUB	26	1A	J	74	4A	u	117	75
ESC	27	1B	K	75	4B	v	118	76
FS	28	1C	L	76	4C	w	119	77
GS	29	1D	M	77	4D	x	120	78
RS	30	1E	N	78	4E	y	121	79
US	31	1F	O	79	4F	z	122	7A
SP	32	20	P	80	50	{	123	7B
!	33	21	Q	81	51	|	124	7C
”	34	22	R	82	52	}	125	7D
#	35	23	S	83	53	~	126	7E
$	36	24	T	84	54	“	127	7F
%	37	25	U	85	55			
&	38	26	V	86	56			
’	39	27	W	87	57			
(	40	28	X	88	58			
)	41	29	Y	89	59			
*	42	2A	Z	90	5A			
+	43	2B						
,	44	2C						
–	45	2D						
.	46	2E						
/	47	2F						

Page laissée blanche intentionnellement

Annexe C Mise en oeuvre CCM Série 90

Le tableau suivant donne la liste des commandes CCM supportées par les API Série 90 :

Tableau C–1. Commandes CCM Série 90

Numéro de la commande	Description
6001	Programmer Réponse Q
6002	Effacer Mots d'état de diagnostic CCM
6003	Lire Mots d'état de diagnostic dans registres source
6004	Configuration logicielle
6101	Lire Table des registres de Destinataire à Source
6102	Lire Table des entrées de Destinataire à Source
6103	Lire Table des sorties de Destinataire à Source
6109	Lire Réponse Q dans Table des registres Source
6110	Ecrire un seul bit
6111	Ecrire Table des registres Source dans Destinataire
6112	Ecrire Table des entrées Source dans Destinataire
6113	Ecrire Table des sorties Source dans Destinataire

Le tableau ci–dessous indique les types de mémoire CCM supportés par les API Série 90 :

Tableau C–2. Types de mémoire CCM supportés par les API Série 90

Type de mémoire CCM	Table CCM destinataire	Opération
1	Table des registres	Lecture/Ecriture
2	Table des entrées	Lecture/Ecriture
3	Table des sorties	Lecture/Ecriture
6	Zone de travail CCM (1)	Lecture
9	Mots d'état de diagnostic (2)	Lecture
13	Table des entrées	Mise de bits à 1
14	Table des sorties	Mise de bits à 1
17	Table des entrées	Mise de bits à 0
18	Table des sorties	Mise de bits à 0

1 Voir dans Tableau 5.3 la configuration de la zone de travail Série 90.
2 Voir dans Tableau 5.4 les mots d'état de diagnostic Série 90 et dans Tableau 5.7 la définition des codes d'erreur Série 90.

Annexe D Mise en oeuvre RTU Série 90

Cette annexe décrit les codes de fonction définis dans les API Série 90.

Tableau D–1. Description des codes de fonction RTU

Codes de fonction	Série 90	Description
0	X	Fonction illégale
1	X	Lire Table des sorties
2	X	Lire Table des entrées
3	X	Lire les registres
4	X	Lire les entrées analogiques
5	X	Forcer une seule sortie
6	X	Prédéfinir un seul registre
7	X	Lire Etat des exceptions
8	X	Bouclage/Maintenance
15	X	Forcer plusieurs sorties
16	X	Prédéfinir plusieurs registres
17	X	Retourner Type d'équipement
67	X	Lire Mémoire de la zone de travail

Page laissée blanche intentionnellement

A

Acronymes, A–2

Adresse de mémoire destinataire (CCM), 5–3

Adresse de mémoire source (CCM), 5–3

Adresses de mémoire, adresses de mémoire CCM, 5–9

Adresses mémoire, adresses mémoire SNP, 6–3

Annuler le datagramme, 7–53

Annuler le datagramme (commande SNP), 6–60

B

BCC, 2–11

Bits d'accès à la mémoire de l'équipement esclave (bits d'état X), 7–57

Bits d'état X, 6–26, 7–57

Bloc de commande de COMREQ, 4–2, 4–7

Bloc de données COMREQ (CCM), 5–2

Bloc de données COMREQ (SNP), 6–2

Bouclage/maintenance (message RTU), 7–35

Bouton de RAZ/Reset, 3–4

Break long, 7–59

C

Câble en "Y" pour CMM311, 3–6

CCM
- adresses de mémoire, 5–9
- exemples de programmation CCM COMREQ, 5–14
- longueurs des données, 5–10
- mot d'état, 5–11
- zone de travail, 5–6

CMM, 1–2

CMM et CCM, 1–2

Code de contrôle de bloc (BCC), 2–11, 7–60

Codes d'erreur
- codes d'erreur du CCM, 5–12
- codes d'erreurs majeures SNP, 6–5
- codes d'erreurs mineures SNP, 6–6
  - codes d'erreur dans un équipement SNP–X esclave, 6–14
  - codes d'erreur du gestionnaire SNP DOS, 6–7
  - codes d'erreur numérotation automatique, 6–13
  - codes d'erreur SNP déportée, 6–12
  - codes d'erreur SNP/SNP–X locale, 6–8
  - codes d'erreurs de demande de service, 6–6

Codes d'erreurs, réponses d'erreurs RTU, 7–43

Codes d'information, 2–4

Codes de type majeur et mineur de l'API, 6–49

Commandes diffusées SNP–X, 7–63

Commandes directes SNP–X, 7–62

Commandes SNP–X, 7–70
- commande Connexion X, 7–70
- commande Ecrire X, 7–75
- commande Lire X, 7–73

Compatibilité (CCM), C–1

Compatibilité (RTU), D–1

Configuration du module CMM, 1–5, 3–12
- configuration du module (configuration logicielle), 3–15
- configuration du rack d'E/S, 3–12

Configuration du module CMM311 avec la miniconsole de programmation HHP, 3–25

Configuration logicielle, 3–15

Configuration logicielle (commande CCM), 5–18

Connexion (commande SNP), 6–32

Connexion longue (commande SNP), 6–63

Contrôle d'erreur par redondance longitudinale (LRC), 2–12

Contrôle de parité, 2–5

Contrôle de redondance cyclique (CRC), 2–13, 7–24
- calcul du CRC–16, 7–25

Convertisseur, 8–8

CRC, 2–13

D

Datagrammes, 7–52
- annuler le datagramme, 6–60, 7–53
- datagramme normal, 6–57, 7–53
- datagramme permanent, 6–57, 7–53
- durée de vie d'un datagramme, 7–53
- établir un datagramme, 6–56, 7–52
- extraction, 7–52
- format Point, 6–57, 7–52
- ID Datagramme, 6–56, 7–52
- mettre à jour le datagramme, 6–59
- mettre à jour le datagramme en temps réel, 6–61
- mise à jour en temps réel, 7–53

Définition des protocoles
- protocole CCM, 7–2
- protocole RTU, 7–20

protocole SNP, 7–46
protocole SNP–X, 7–55

Définition des protocoles – CCM, RTU, SNP, SNP–X, 7–1

Délai de diffusion, 6–29, 6–31, 7–58

Délai de retournement du modem, 7–51

Délai de transmission, 6–29, 6–31, 7–51

Demande de communication (COMREQ), 4–2

Demande et réponse SNP, 7–48

Description d'une communication avec le module CMM, 2–3

Description matérielle, 3–2

Diffusion des ID SNP, 7–56

E

Ecrire dans l'équipement destinataire depuis l'équipement source (commande CCM), 5–22

Ecrire dans mémoire des blocs du programme (commande SNP), 6–43

Ecrire dans mémoire des tâches (commande SNP), 6–41

Ecrire dans mémoire système (commande SNP), 6–39

Ecrire un seul bit (commande CCM), 5–21

Ecrire X (commande SNP), 6–30

Effacer les mots d'état de diagnostic CCM (commande CCM), 5–17

Effacer mots d'état de diagnostic (commande SNP), 6–22

Egal à égal, 7–7

Erreurs de bruit, 2–5

Erreurs de cadence, 2–6

Erreurs de synchronisation, 2–6

Erreurs de temporisation, 2–6

Erreurs de trame, 2–6

Erreurs de transmission et détection, 2–5

Essais de retransmission (SNP), 7–49

Etablir un datagramme (commande SNP), 6–56, 7–52

Etablissement d'une communication SNP, 7–47

Etablissement d'une session SNP–X, 7–62

Etat complémentaire, 6–33, 6–34, 6–63
mot d'état de l'API, 6–35

Etat court, 6–44

Etat court de l'API (commande SNP), 6–44

Exemple de programme à diagramme en échelle, exemple de programme SNP, 6–17

Exemple de programme en diagramme en échelle, exemple de programmation CCM, 5–14

Exemples de commandes CCM, 5–16
commande Programmer la réponse Q, 5–16
configuration logicielle, 5–18
écrire dans l'équipement destinataire depuis l'équipement source, 5–22
écrire un seul bit, 5–21
effacer les mots d'état de diagnostic CCM, 5–17
lire la mémoire de l'équipement destinataire dans la mémoire de l'équipement source, 5–19
lire la réponse Q dans la table des registres de l'équipement source, 5–20
lire les mots d'état de diagnostic CCM dans les registres source, 5–17

Exemples de commandes SNP, 6–23
annuler le datagramme, 6–60
connexion, 6–32
connexion longue, 6–63
écrire dans mémoire des blocs du programme, 6–43
écrire dans mémoire des tâches, 6–41
écrire dans mémoire système, 6–39
écrire X, 6–30
effacer mots d'état de diagnostic, 6–22
établir un datagramme, 6–56
état court de l'API, 6–44
forcer bascule de la mémoire système, 6–54
lire les mots d'état de diagnostic, 6–23
lire mémoire des blocs du programme, 6–42
lire mémoire des tâches, 6–40
lire mémoire système, 6–38
lire X, 6–28
mettre à jour le datagramme, 6–59
mettre à jour le datagramme en temps réel, 6–61
modifier ID SNP, 6–25
modifier le niveau de protection, 6–36
numérotation automatique, 6–66
programmer adresse des bits d'état X, 6–26
programmer Heure/Date de l'API, 6–53
retourner Heure/Date de l'API, 6–50
retourner nom du programme de commande, 6–46
retourner table des défauts, 6–51
retourner Type et ID du contrôleur, 6–48

Exemples de programmation
exemple de programme SNP, 6–17
exemples de programmation CCM, 5–14

Exemples de programmation de SNP COMREQ, 6–17

Exemples de programmation SNP–X COMREQ, 6–21
écrire X, 6–30
lire X, 6–28

programmer adresse des bits d'état X, 6–26

Extraction d'un datagramme, 7–52

F

Fonctionnement système du module CMM, 3–7

Forcer bascule de la mémoire système (commande SNP), 6–54

Forcer plusieurs sorties (message RTU), 7–37

Forcer une seule sortie (message RTU), 7–32

Format de la table des défauts, 6–52

Format des messages SNP, 7–47

Format Point, 6–57
restrictions des types de mémoire, 6–58

G

Glossaire, A–1

Guide de référence du manuel, 1–1

I

ID CPU (CCM), 5–2

ID destinataire (CCM), 5–2

ID SNP, 6–25, 6–29, 6–31, 6–33, 6–61, 6–64, 7–47, 7–56

Installation du module CMM, 3–10

Installation et configuration du module CMM, 3–9

Instruction COMREQ, 4–2, 4–5

Interface d'application de l'API, 2–14

Interface RS–485, 8–4
fonctionnement sur 2 fils, 5–18

Interface série, 2–4

Interfaces normalisées, 2–8

Interrogation–réponse, 7–20

Introduction aux modules coprocesseur de communication (CMM), 1–2

L

Liaison série, 2–7

Lire état des exceptions (message RTU), 7–34

Lire la mémoire de l'équipement destinataire dans la mémoire de l'équipement source (commande CCM), 5–19

Lire la réponse Q dans la table des registres de l'équipement source (commande CCM), 5–20

Lire les entrées analogiques (message RTU), 7–31

Lire les mots d'état de diagnostic (commande SNP), 6–23

Lire les mots d'état de diagnostic CCM dans les registres source (commande CCM), 5–17

Lire les registres (message RTU), 7–30

Lire mémoire de la zone de travail (message RTU), 7–40

Lire mémoire des blocs du programme (commande SNP), 6–42

Lire mémoire des tâches (commande SNP), 6–40

Lire mémoire système (commande SNP), 6–38

Lire table des entrées (message RTU), 7–29

Lire table des sorties (message RTU), 7–28

Lire X (commande SNP), 6–28

Liste des codes ASCII, B–1

Longueur des données (CCM COMREQ), 5–3

Longueur des données (CCM), 5–3

Longueurs des données (Types de mémoires CCM), 5–10

LRC, 2–12

M

Maître–esclave, 7–9, 7–46

Mettre à jour le datagramme (commande SNP), 6–59

Mettre à jour le datagramme en temps réel (commande SNP), 6–61

Miniconsole de programmation (HHP), 3–25

Mise à jour du datagramme en temps réel, 7–53

Mode CCM d'égal à égal, 7–7

Mode CCM maître–esclave, 7–9

Mode NOWAIT, 4–7

Mode WAIT, 4–7

Modèle de références des communications, 2–2

Modems, 2–7

Modes de communication par modem, 2–7

Modes de configuration, 3–15
mode CCM seul, 3–16

mode CCM/RTU, 3–16
mode CCM/SNP, 3–19
mode RTU seul, 3–17
mode RTU/CCM, 3–17
mode RTU/SNP, 3–20
mode SNP seul, 3–18
mode SNP/CCM, 3–18
mode SNP/RTU, 3–19
Modifier ID SNP (commande SNP), 6–25
Modifier le niveau de protection (commande SNP), 6–36
Mono–session, 6–29, 6–31
Mot d'état
mot d'état CCM, 5–11
mot d'état SNP, 6–4
Mot d'état COMREQ
mot d'état CCM, 5–11
mot d'état SNP, 6–4
Mot d'état de l'API, 6–35
Mot d'état de l'API esclave (SNP–X), 6–29, 6–31, 7–57
Mot d'état SNP, 6–4
Mots d'état de diagnostic (CCM), 5–8
Mots d'état de diagnostic (SNP), 6–23
Multi–session, 6–29, 6–31
Multipoint, 2–16

N

Niveaux de protection de l'API, 6–37
Nombre d'essais (CCM), 7–13
Numéro de la commande CCM, 5–2
Numérotation automatique (commande SNP), 6–66

P

Paramètres de communication, 3–21
Partie matérielle du module CMM, 1–4
Point–à–point, 2–15
Ports série, 3–4
CMM311, 3–5
CMM711, 3–4
Prédéfinir plusieurs registres (message RTU), 7–38
Prédéfinir un seul registre (message RTU), 7–33
Présentation du module CMM, 1–2
Principe des communications CMM, 2–2
Principe du mode SNP maître–esclave, 7–46
Programmer adresse des bits d'état X, 6–26
Programmer Heure/Date de l'API (commande SNP), 6–53
Programmer la réponse Q (commande CCM), 5–16
Protocole CCM, 2–11, 7–2
synchronisation et nombre d'essais, 7–11
tables des états, 7–13
Protocole RTU, 2–12, 7–20
Protocole SNP, 7–46
Protocole SNP–X, 7–55
Protocoles de communication (SNP, SNP–X, CCM, RTU), 2–10
Protocoles de communication du module CMM, 1–5
Protocoles SNP et SNP–X, 2–10

R

Répéteur/convertisseur, 8–8

Répéteur/convertisseur isolé, 8–8
  configurations système, 8–12
  schémas des câbles, 8–14

Réseaux de communication, 2–15

Résumé des commandes CCM, 5–4

Résumé des commandes SNP, 6–21

Retourner Heure/Date de l'API (commande SNP), 6–50

Retourner nom du programme de commande (commande SNP), 6–46

Retourner table des défauts (commande SNP), 6–51

Retourner type d'équipement (message RTU), 7–39

Retourner Type et ID du contrôleur (commande SNP), 6–48

RS–232, 2–8

RS–449, RS–422 et RS–485, 2–9

RTS/CTS, 3–23

RTU
  champs de messages, 7–21
    adresse de la station, 7–22
    champ contrôle d'erreurs, 7–23
    champ Informations, 7–22
    code de la fonction, 7–22
  compatibilité, D–1
  description des messages, 7–28
    bouclage/maintenance, 7–35
    forcer plusieurs sorties, 7–37
    forcer une seule sortie, 7–32
    lire état des exceptions, 7–34
    lire les entrées analogiques, 7–31
    lire les registres, 7–30
    lire mémoire de la zone de travail, 7–40
    lire table des entrées, 7–29
    lire table des sorties, 7–28
    prédéfinir plusieurs registres, 7–38
    prédéfinir un seul registre, 7–33
    retourner type d'équipement, 7–39
  erreurs de communication, 7–43
  fin du message, 7–23
  format des caractères, 7–23
  format des messages, 7–20
  longueur des messages, 7–27
  types de messages, 7–21
    diffusion, 7–21
    interrogation, 7–21
    réponse d'erreur, 7–21
    réponse normale, 7–21
  utilisation des temporisations, 7–23
  zone de travail, 7–41

S

Schémas de câblage, 8–1

Schémas de câblage pour répéteur/convertisseur isolé, 8–14

Schémas de câblage RS–232, 8–3

Schémas de câblage RS–422/RS–485, 8–4

Schémas de câblage série, 8–1

Séquence Break, 7–47

Séquence du protocole SNP–X, 7–61

Séquence Q, 7–10

Service CCM, 5–1

Service SNP, 6–1

Session
  session de communication SNP, 7–47
  session de communication SNP–X, 7–55

SNP
  essais de retransmission et reprise sur incident, 7–49
  temporisateurs, 7–49

Spécifications d'assemblage du câble, 8–2

Structure des messages de demande X, 7–65

Structure des messages réponse X, 7–67

Structure du message tampon X, 7–69

Synchroniser Date/heure API, 6–53

T

Tables des états (CCM), 7–13
  table des états CCM esclave, 7–18
  table des états CCM homologue, 7–14
  table des états CCM Maître, 7–16

Tables des états SNP–X, 7–82
  table des états de l'équipement SNP–X esclave, 7–84
  table des états de l'équipement SNP–X maître, 7–82

Temporisateur (SNP–X), 7–58
  délai de diffusion, 7–58
  temporisation de la réponse, 7–58
  temporisation du tampon, 7–59

Temporisateurs (SNP), temporisations SNP par défaut, 3–24

Temporisateurs SNP, 7–49
  T1, 7–49
  T2, 7–50
  T3, 7–50
  T3”, 7–50
  T3’, 7–50
  T4, 7–50
  T5, 7–51
  T5”, 7–51
  T5’, 7–51

Temporisation de la réponse, 7–58

Temporisations (CCM), 7–11

Temporisations (RTU), 7–23, 7–44

Temps de retournement du modem, 6–29, 6–31

Temps limite de réponse, 6–29, 6–31

Transmission asynchrone, 2–6

Transmission par modem, 2–17

Type de mémoire destinataire (CCM), 5–3

Type de session (SNP–X), 6–29, 6–31

Types de mémoires
  types de mémoires CCM, 5–5, C–2
  types de mémoires SNP, 6–3

Types de mémoires CCM, 5–5

V

Voyants, 3–3
  voyant Module OK, 3–3
  voyants Ports série, 3–3

Z

Zone de travail (CCM), 5–6

Zone de travail RTU, 7–41

Zone informations, 6–49

Zone programmes de commande, 6–47